Terascale Optimal PDE Simulations (TOPS), An Enabling Technology Center

Scientific Discovery Through Advanced Computing:
Integrated Software Infrastructure Centers

Proposing Organization:
Old Dominion University

David E. Keyes, Principal Investigator
Mathematics & Statistics Department
Old Dominion University
Norfolk, Virginia 23529-0077
Phone 757-683-3906
Fax 757-683-3885
dkeyes@odu.edu

Annemarie Delgado, Authorizing Official
Old Dominion University Research Foundation
P.O. Box 6369
Norfolk, Virginia 23508-0369
Phone 757-683-5685
Fax 757-683-5920
adelgado@odu.edu

Requested Funding by Fiscal Year:
2001: $3,976,733 *2002:* $4,473,040 *2003:* $4,987,611 *2004:* $5,164,771 *2005:* $5,377,892
Total Requested Funding:
2001–2005: $23,980,047

No use of human subjects in proposed project
No use of vertebrate animals in proposed project

David E. Keyes

Annemarie Delgado

Contents

Abstract **iii**

1 Background and Significance **1**

2 Preliminary Studies **3**
2.1 PDE Time Integrators . . . 3
2.2 PDE Nonlinear Solvers . . . 4
2.3 PDE-constrained Optimization . . . 4
2.4 Linear Solvers . . . 5
2.5 Eigensystem Analysis . . . 6
2.6 Integration and Deployment . . . 7
2.7 Software Performance Tuning . . . 7

3 Research Design and Methods **8**
3.1 PDE Time Integrators . . . 8
3.2 PDE Nonlinear Solvers . . . 8
3.2.1 Schwarz Preconditioned Inexact Newton . . . 9
3.2.2 Newton-Krylov-Multigrid . . . 9
3.2.3 Nonlinear Multigrid (FAS) . . . 9
3.2.4 Common Framework for Three Complementary Approaches . . . 10
3.3 PDE-constrained Optimization . . . 10
3.3.1 Optimization Methods . . . 11
3.3.2 Sensitivity Analysis . . . 11
3.4 PDE Linear Systems . . . 12
3.4.1 Algebraic Multigrid Methods . . . 12
3.4.2 Geometric Multigrid Methods . . . 14
3.4.3 ILU Preconditioners . . . 15
3.4.4 Sparse Direct Methods . . . 16
3.5 PDE Eigenanalysis . . . 17
3.6 PDE Software Integration and Deployment . . . 19
3.7 PDE Software Performance . . . 20
3.7.1 Automated Empirical Optimization of Software (AEOS) . . . 20
3.7.2 Problem-dependent Tuning . . . 21
3.8 Milestones . . . 21

4 Consortium Arrangements **24**
4.1 Other SciDAC ETCs . . . 24
4.2 SciDAC Application Groups . . . 24
4.3 Internal Collaboration . . . 24

References **26**

5 Budget **35**

6 Other Support **36**

7 Biographical Sketches **37**

8 Facilities and Resources **64**
8.1 LBNL/NERSC 64
8.2 ANL 64
8.3 LLNL 64
8.4 Carnegie Mellon University 65
8.5 University of Colorado-Boulder 65
8.6 Old Dominion University 65

9 Appendices **66**
9.1 Other Active Participants 66
9.2 Relevant Publications 66

Abstract

Large-scale simulations of importance to the Department of Energy often involve the solution of partial differential equations (PDEs). In such simulations, continuous (infinite-dimensional) mathematical models are approximated with finite-dimensional models. To obtain the required accuracy and resolve the multiple scales of the underlying physics, the finite-dimensional models must often be extremely large, thus requiring terascale computers. Fortunately, continuous problems provide a natural way to generate a hierarchy of approximate models, through which the required solution may be obtained efficiently by various forms of "bootstrapping." The most dramatic examples are multigrid methods, but we also exploit other hierarchical representations herein.

We propose an Enabling Technology Center (ETC) that focuses on developing, implementing, and supporting optimal or near optimal schemes for PDE simulations and closely related tasks, including optimization of PDE-constrained systems, eigenanalysis, and adaptive time integration. The Terascale Optimal PDE Simulations (TOPS) Center will research, develop, and deploy an integrated toolkit of open source, (nearly) optimal complexity solvers for the nonlinear partial differential equations that arise in many Office of Science application areas, including fusion, accelerator design, global climate change, and reactive chemistry. These algorithms — primarily multilevel methods — aim to reduce computational bottlenecks by one to three orders of magnitude on terascale computers, enabling scientific simulation on a scale heretofore impossible. Along with usability, robustness, and algorithmic efficiency, an important goal of this ETC will be to attain the highest possible computational performance in its implementations by accommodating to the memory bandwidth limitations of hierarchical memory architectures.

1 Background and Significance

Multicomponent nonlinear partial differential equations (PDEs) provide the common mathematical expression of many DOE simulations, including accelerator design, fusion, diesel and internal combustion engines, groundwater and oil reservoirs, and climate. Such simulation codes require implicit solvers for multiscale, multiphase, multiphysics phenomena from hydrodynamics, electromagnetism, radiation transport, chemical kinetics, and quantum chemistry. Problem sizes are typically now in the millions of unknowns; and with emerging large-scale computing systems and inexpensive clusters, we expect this size to increase by a factor of a thousand over the life of the project. Moreover, these simulations are increasingly used for design optimization, parameter identification, and process control applications that require many repeated, related simulations.

Unfortunately, the implicit solution algorithms currently used in many Office of Science applications have far from optimal computational complexities and are invariably bottlenecks that limit the scalability of the entire application, independent of the quality of the implementation. For example, an increase in problem size of a factor of 100 can easily result in an increase in work requirements of 1000. In comparison, optimal complexity algorithms have work (and memory) requirements that grow only linearly with problem size. Multilevel (or multigrid) methods make up a class of optimal complexity algorithms that have produced spectacular improvements in overall simulation time (for example, we have demonstrated speedups of up to two orders of magnitude in solver time in some legacy DOE groundwater and fusion simulations). However, current multilevel software tends to be problem-specific and is mature only for scalar (as opposed to multicomponent) PDEs. Because of the potential payoff, this ETC will expend much of its effort on developing practical, usable multilevel methods for comprehensive aspects of PDE simulations.

This ETC is concerned with five PDE simulation capabilities: adaptive time integrators for stiff systems, nonlinear implicit solvers, optimization, linear solvers, and eigenanalysis. The relationship between these areas is depicted in Figure 1. In addition, the ETC will contain two cross-cutting topics: software integration (or interoperability) and high-performance coding techniques for PDE applications.

Optimal (and nearly optimal) complexity numerical algorithms almost invariably depend upon a hierarchy of approximations to "bootstrap" to the required highly accurate final solution. Generally, an underlying continuum (infinite-dimensional) high fidelity mathematical model of the physics is discretized to "high" order on a "fine" mesh to define the top level of the hierarchy of approximations. The representations of the problem at lower levels of the hierarchy may employ other models (possibly of lower physical fidelity), coarser meshes, lower order discretization schemes, inexact linearizations, and even lower floating-point precisions. The philosophy that underlies our algorithmics and software is to make the majority of progress towards the highly resolved result through possibly low-resolution stages that run well on high-end distributed hierarchical memory computers.

The ingredients for constructing hierarchy-of-approximations-based methods are remarkably similar, be it for solving linear systems, nonlinear problems, eigenvalue problems, or optimization problems, namely:

1. A method for generating several discrete problems at different resolutions (for example on several grids)
2. An inexpensive method for iteratively improving an approximate solution at a particular resolution
3. A means of interpolating (discrete) functions at a particular resolution to the next finer resolution
4. A means of transferring (discrete) functions at a particular resolution to the next coarser resolution (often obtained trivially from interpolation).

We believe that software should reflect the simplicity and uniformity of these ingredients over the five problem classes and over a wide range of applications. We expect that with experience we will achieve a

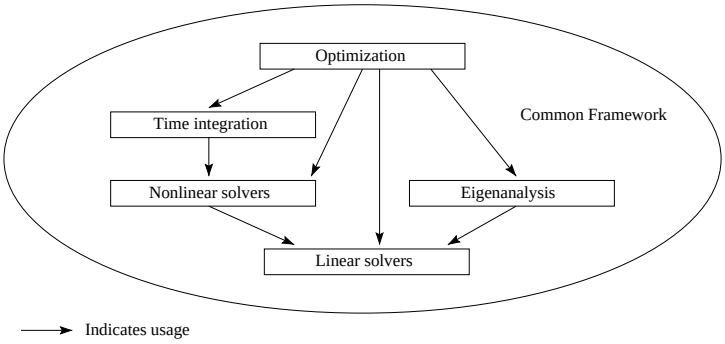

Figure 1: *An arrow from A to B indicates that A typically uses B. Optimization of systems governed by PDEs requires repeated access to a PDE solver. The PDE system may be steady-state or time-dependent. Time-dependent PDEs are typically solved with implicit temporal differencing. After choice of the time-integration scheme, they, in turn, require the same types of nonlinear solvers that are used to solve steady-state PDEs. All algorithms for nonlinear problems of high dimension generate a sequence of linear problems, so linear solver capability is at the core. Eigenanalysis arises inside of or independently of optimization. Like direct PDE analysis, eigenanalysis generally depends upon solving a sequence of linear problems. All of these five classes of problems, in a PDE context, share grid-based data structures and considerable parallel software infrastructure. Therefore, it is compelling to undertake them together.*

reduction in the number of lines of code that need to be written and maintained, because the same code can be reused in many circumstances.

Algorithms and software for the solution of linear and nonlinear systems of equations, especially those arising from PDEs, have been principal emphases of the Department of Energy research portfolio for decades. This points both to the central importance of this project, and also to the historical difficulty of reconciling the conflicting objectives of solver software technology. Solvers are supposed to be of *general purpose*, since a great diversity of applications require them, but they are also supposed to be *highly performant*, since they are often the inner loops of such applications. However, high performance usually requires exploitation of special structure (e.g., symmetry, dense blocking, geometrical or coefficient regularity), which may be different in different applications. Then, too, solvers for PDEs are supposed to be *robust* across all regimes of use, since scientists trained in the application domain cannot also be required to be expert in tuning solvers, but they are also supposed to have *optimal complexity*, since desired discrete problem size is limited only by the validity of the continuum model. Once again, algorithmic optimality (work and memory requirements a small multiple of their information-theoretic minima) is generally achieved by exploitation of special structure that cannot be assumed in a robust code.

These conflicting objectives do not describe a hopeless situation, however. The opportunity for 21st century solver developers is to exploit advances in object-oriented programming to construct highly versatile and adaptive software that finds, creates, and exploits structure wherever possible, while automatically "falling back" to conservative approaches in the remaining (hopefully lower-dimensional) parts of a problem. The solver toolkit of the future will be a collection of objects with rich and recursive interconnections, rather than a collection of subroutines through which a relatively small number of calling sequences are predefined. Algorithmic theory, scientific software engineering, and understanding of architecturally-motivated performance optimizations have all advanced significantly since the last time many applications communities "fastened onto" their canonical solver technology. Advances along these three fronts must be packaged, refined, freshly promoted, and supported for the benefit of the user community.

The efforts defined in this proposal, the co-PIs joining to undertake them, and the alliances proposed

with other groups have been carefully chosen to exploit the present opportunity to revolutionize large-scale solver infrastructure, and lift the capabilities of dozens of DOE's computational science groups as an outcome. Included among the co-PIs are several of the world's leading theorists in multilevel/hierarchical methods — continuous *and* discrete, a "Who's Who" of deliverers of the current generation of freely available portable parallel solver libraries, and a group of interdisciplinary computational scientists that have already begun to transform the computing practices of specialized user communities (such as computational aerodynamics, computational radiation transport, and computational reservoir modeling) for high-performance production parallel simulation.

The co-PIs have extensive networks of previous collaborations with each other and with key co-PIs in other groups with which natural development synergisms exist — including complex geometry and adaptive gridding, high performance, common component architecture, and visualization. They also have two tight collaborations planned with applications teams — the Princeton-based and LANL-based magnetic fusion energy group and the ORNL-based astrophysics group. These primary collaborations will be allowed to influence TOPS priorities for software development. Of course, since we intend to publish the next generation of some solver software that already has user groups today, we expect considerable additional user interaction beyond these primary collaborators.

Just as we expect our user community to drive our research and development work, we expect to significantly impact the scientific priorities of our user group by emphasizing optimization (inverse problems, optimal control, optimal design) as part of our solver toolkit. As explained below and in the references, optimization subject to PDE-constraints is a particularly active subfield of optimization because the traditional means of handling constraints in black-box optimization codes — with a call to a PDE solver in the inner loop — is too expensive. We are emphasizing "simultaneous analysis and design" methods in which the cost of doing the optimization is a small multiple of doing the simulation and the simulation data structures are actually part of the optimization data structures.

Our motivating belief is that most PDE simulation is ultimately a part of some larger scientific process that can be hosted by the same data structures and carried out with many of the same optimized kernels as the simulation, itself. We intend to make the connection to such processes explicit and inviting to users, and this will be a prime metric of our success. The assembly of the project team and the organization of the proposal flow directly from this program: *Terascale software for PDEs should extend from the analysis to the scientifically important auxiliary processes of sensitivity analysis, modal analysis (e.g., of perturbations about the PDE solution), and the ultimate "prize" of optimization subject to conservation laws embodied by the PDE system.*

2 Preliminary Studies

The proposed research in Section 3 builds on the experience and, in many cases, existing codes of the co-PIs, in seven areas. This section highlights a diverse selection of accomplishments on which Section 3 builds. With the exception of §2.1, one of the co-PIs led all of the preliminary studies described herein.

2.1 PDE Time Integrators

General-purpose software packages previously developed at LLNL (apart from the co-PIs) for integration of stiff ODEs and DAEs are among the most widely used solvers anywhere for these problems, and through the method of lines, for time-dependent PDEs, as well. One predecessor code to PVODE, LLNL's most recent parallel ODE solver, has almost 270,000 downloads from the netlib mathematical software library. PVODE has been interfaced with the Portable Extensible Toolkit for Scientific Computing library (PETSc; see

http://www.mcs.anl.gov/petsc/ and [12, 13]). This project proposes to support LLNL's PVODE [45] and IDA [87] software for parallel method-of-lines applications through PETSc and other frameworks.

IDA has been effectively used to solve a 9 billion unknown time-dependent Boltzmann transport equation on over 4,000 processors of the ASCI White machine. PVODE has been incorporated into a 3D edge-plasma turbulence model called BOUT, for which superlinear parallel speedups were measured on 60 processors of a Cray T3E. PVODE has also been used in a radiation diffusion code on over 5,800 processors of the ASCI Red machine.

2.2 PDE Nonlinear Solvers

The co-PIs bring to this project a strong base of algorithmic research and existing parallel software for solution of large-scale discretizations of systems of PDEs. Newton-Krylov-Schwarz (NKS) methods [46] and Newton-Krylov-multigrid methods [137] have proven to be broadly applicable, architecturally versatile, and tunable for high performance on today's high-end commercial parallel platforms (e.g., Cray T3E, SGI Origin, IBM SP, Compaq AlphaServer) and also on modest-sized beowulfs.

The crux of Newton-Krylov-Schwarz (NKS) algorithms is a balance between implicit convergence rate and good data locality [98]. Work is organized by subdomains to accommodate the pronounced memory hierarchies of modern high-performance machines, in which both vertical (memory/cache) and horizontal (processor/processor) transfer of data exacts a performance price. Global operations supplement the subdomain work to be nearly algorithmically scalable while preserving implementation scalability [97]. Newton-Krylov-multigrid methods make use of the optimal scaling properties of multigrid for preconditioning and the fast, quadratic convergence of Newton's method for the nonlinear solve with the result of a scalable, fast nonlinear solver. Newton's method, robustified by pseudo-transient continuation [96], generates global linear systems that are solved in a matrix-free manner by Krylov iteration, and preconditioned locally with inexact linear solvers or even nonlinear solvers on subsystems [47]. Matrix-free NKS is one of the prime existing nonlinear solvers in the PETSc library.

Matrix-free Newton-Krylov is the heart of the Krylov Inexact Newton SOLve package (KINSOL [130], see http://www.llnl.gov/CASC/PVODE/), a parallel C code developed from the original Newton-Krylov code, NKSOL [43].

Both structured-grid [79] and unstructured-grid [78] CFD legacy codes have been ported to advanced platforms using PETSc, with excellent algorithmic convergence rate, parallel efficiency, and raw floating point performance. A philosophy key to our success is the use of multiple representations of a problem—a highly accurate one by which to drive and in which to measure convergence progress and a hierarchy of approximate representations for low storage and fast preconditioning "inversion". Our implementation of the unstructured grid NASA code FUN3D running on up to 6,144 processors of the ASCI Red machine at 227 Gigaflop/s—won a 1999 Bell Prize [6].

Use of the KINSOL code has enabled the expansion into new physics for LLNL hydrologists. A recent push for variably saturated subsurface flow modeling resulted in the need for solvers for large-scale nonlinear parabolic equations. We employed the KINSOL package with a structured multigrid preconditioner from the HYPRE library for this model. This solver proved effective and enabled the advance from linear saturated flow models to nonlinear variably saturated models [106].

2.3 PDE-constrained Optimization

The co-PIs include leaders in the design and implementation of algorithms for large-scale optimization. Our combined expertise provides the required background for attacking PDE-constrained problems of interest to the Office of Science.

We have developed optimization algorithms that offer the promise of solving the PDE-constrained optimization problem at a cost that is a small multiple of the cost of a single nonlinear PDE solve [25, 26, 27, 28].

These methods have been applied to optimization problems constrained by Navier-Stokes [7, 28, 71, 72, 82], nonlinear elasticity [115], acoustic wave propagation [2], inviscid compressible flows [114], and coupled nonlinear flow-solid [73] simulations. The most recent variants of our *Lagrange–Newton–Krylov–Schur* (LNKS) optimization algorithms have been shown to scale well with increasing numbers of state variables, optimization variables, and processors. For example, we were able to solve an optimal boundary control problem for Navier-Stokes flow involving 10^6 state and 10^4 control variables in just 5 times the cost of a single Navier-Stokes solve [28], with high parallel efficiency on 256 Cray T3E processors. These solution times are 1–2 orders of magnitude smaller than those taken by contemporary optimization methods. The LNKS family of algorithms has been incorporated into VELTISTO [24, 23], a software library for parallel PDE-constrained optimization. VELTISTO is built on PETSc and is currently being hardened for a public release later this year.

We have also developed the Toolkit for Advanced Optimization (TAO) [20] under the DOE2000 program. TAO focuses on the design and implementation of component-based optimization software for the solution of large-scale optimization applications. Our design enables connection to lower-level support (parallel sparse matrix data structures, preconditioners, solvers) provided in toolkits such as PETSc and thus we are able to build on top of these toolkits instead of having to redevelop code. The advantages in terms of development time are significant. Version 1.0 of TAO (see `http://www.mcs.anl.gov/tao`), released in October 2000, runs on a wide variety of high-performance architectures. Our benchmarking results [19] for bound-constrained codes are noteworthy because we are able to solve optimization problems with over 2.5 million variables with only 8 processors and between 80% and 90% efficiency. This sustained efficiency is unexpected in classical active set methods.

We also have wide experience with the use of automatic differentiation tools in optimization software. Of particular interest is NEOS [57, 76] and recent work on integrating automatic differentiation with object-oriented toolkits [1], and on using automatic differentiation for second-order methods in PDE-constrained optimization [88].

2.4 Linear Solvers

The co-PIs are among the world's leaders in multilevel methods research. They were significant contributors in the early stages of development of the algebraic multigrid (AMG) method [39, 40, 123], and continue to make important contributions to the field today through new algorithmic innovations such as AMGe [41, 54, 55, 86, 93]. The co-PIs also bring experience in the area of geometric multigrid methods for problems discretized on structured and semi-structured grids [8, 14, 42], such as the composite-grid problems used in structured adaptive mesh refinement (SAMR) applications [107], a focus of this ETC. This ETC will also leverage extensive experience with parallel multigrid methods for large-scale scientific computing [8, 42, 54, 69]. These methods have demonstrated excellent scalability on up to 3150 processors of ASCI Red. The co-PIs already enjoy an existing close collaboration in multigrid methods research, and as such, are ideal for this ETC.

The majority of the multilevel algorithms mentioned above either have been or will be implemented in HYPRE, a library of high-performance parallel preconditioners (`http://www.llnl.gov/CASC/hypre/`). Through libraries such as HYPRE, the co-PIs have moved much of their fundamental multilevel methods research into parallel application codes, impacting such diverse areas as groundwater and water resource management, radiation hydrodynamics, laser plasma interaction, and structural mechanics. In inertial confinement fusion (ICF) applications, early sequential studies demonstrated speedups in overall simulation time by up to a factor of ten using multigrid [14]. This result has since been demonstrated in parallel ICF codes with speedups of up to a factor of six. For groundwater flow applications, the impact has proven even more dramatic with speedups in linear solve times of up to a factor of 100 [8]. For parallel adaptive laser plasma simulations, HYPRE solvers have allowed simulations that could not previously be done.

The PETSc toolkit contains a very rich set of Krylov solvers (over 12) as well as a variety of parallel

preconditioners and interfaces to other packages. These include the additive Schwarz method, multigrid, the balancing Neumann-Neumann method, SPAI and ILU(0)/ICC(0). They are all accessed via a common interface thus allowing the user to select the most appropriate solver for his or her class of problem at runtime.

The PETSc linear solvers have been used for wide classes of problems, including Euler and Navier-Stokes fluid flow, particulant flow, structural mechanics, plasticity, EMF, econometrics, acoustics, multiphase flow, and sintering; see `http://www-unix.mcs.anl.gov/petsc/docs/petscapps.html`. For example, the largest time-dependent, fully coupled compositional (oil) reservoir simulations with over 32 million degrees of freedom have been run using PETSc solvers.

In the area of preconditioners, the co-PIs developed a scalable parallel algorithm for computing incomplete factorization, using a level-of-fill approach, which was a runner-up for "Best Paper" at Supercomputing'99. It creates parallelism by graph partitioning and preserves the concurrency by ordering the subgraphs and the computations within each subgraph in order to limit fill elements. Scalability has been demonstrated by analysis and by an implementation on up to 216 processors using problems with 20 million unknowns. Preliminary numerical experiments show that the number of preconditioned iterations for a specified residual reduction increases only very slowly with the number of processors, as the problem is partitioned into increasing numbers of subgraphs [89].

Many parallel ILU preconditioners use sequential ILU algorithms on individual processors. The co-PI's recent studies show that there is a lot to be learned even about these workhorse sequential value-based ILU algorithms. The amount of fill needed for convergence can be quite large, thus increasing the need for memory-efficient codes. One simple improvement allows more non-zeros to be kept in later rows of the factors, which are likely to be denser. Another improvement of greater potential benefit involves first permuting the matrix into block upper triangular form and working on the subproblems individually. Tests show many problems benefit from this ordering.

The co-PIs have extensive experience in developing sparse direct solvers. For sparse symmetric positive definite matrices, we have provided efficient tuned versions of a left-looking column block solver PANFCT [111, 113], a multifrontal solver DSCPACK [120], and a currently sequential object-oriented sparse solver OBLIO [67]. DSCPACK is suitable for certain classes of indefinite matrices and provides limited forms of pivoting numerical stability, while OBLIO can handle both symmetric positive definite and indefinite problems. PANFCT and DSCPACK are designed for shared-memory and distributed-memory, respectively.

For sparse nonsymmetric matrices, we have developed a comprehensive set of routines, packaged in SuperLU library [61]. The package contains a number of novel performance-enhancing techniques that lead to high speed and scalability on modern computer architectures [59, 60, 101]. GESP is a special version of SuperLU that uses static pivoting instead of partial pivoting. A variety of techniques are employed in GESP to retain numerical stability. Static pivoting effectively avoids much of the overhead that would be incurred when pivoting were performed. It also enables static optimizations of data distribution and communication.

The solvers mentioned above are widely used in scientific and engineering applications, which include computational fluid dynamics, structural analysis, ground water simulations, circuit and device simulations, and numerical optimization. Both BLKFCT and SuperLU have been incorporated in many mathematical libraries, such as Matlab, Sun Microsystems, HP, Convex, and Boeing BCSLIB-EXT. The MPP version of SuperLU was crucial in the solution of a 50-year-old computational chemistry problem of electron impact ionization with three charged particles; SuperLU was used to solve linear systems of dimensions up to 1.8 million. This significant result made the cover of the journal *Science*, Dec. 24, 1999. More details can be found in [11].

2.5 Eigensystems Analysis

The kernel of any sparse or dense symmetric eigensolver, or singular value decomposition algorithm, is the solution of a symmetric tridiagonal eigenproblem (or bidiagonal SVD). This is the parallel bottleneck when

the traditional QR algorithm is used. In joint work with B. Parlett and I. Dhillon [63, 64, 65, 70, 116], we have been developing what we call the "Holy Grail" of such solvers: (1) it costs just $O(kn)$ work to compute all kn components of k n-dimensional eigenvectors, i.e. it has optimal complexity; (2) each eigenvector can be computed in an embarrassingly parallel fashion; and (3) it has guaranteed accuracy, i.e. the eigenvectors are orthogonal. A preliminary version of this routine has been released in LAPACK (see [5, Figures 3.1 and 3.2] or `http://www.netlib.org/lapack/lug/node71.html`).

2.6 Integration and Deployment

The PIs on this proposal have had major roles in the development of some of the most widely used linear algebra and optimization software libraries. These include LINPACK, MINPACK, MINPACK-2, LAPACK, ScaLAPACK, PETSc, and SuperLU. Several of these packages have been licensed by commercial companies and included in products such as Matlab and vendors' numerical software libraries. All of them have been continuously supported as Open Source, one for more than 20 years. Another project worth mentioning is the long-term ODE time-integrators work at LLNL whose current parallel instantiation is PVODE.

In addition to the software library development, the PIs have played major roles in developing and promoting interoperability between numerical software libraries. For example, as part of DOE 2000 work, interfaces between the ANL PETSc solvers and two LLNL parallel grid packages, Overture and SAMRAI, as well as the ODE package PVODE, have been developed. Long before DOE 2000 funding, PETSc was already using portions of LAPACK, LINPACK, BlockSolve95, SPARSEPAK, MINPACK, ParMeTiS, and SPAI to eliminate redundant code development. Recently LLNL has demonstrated a (pre-release) version of the HYPRE preconditioner suite that provides support for the use of the PETSc matrix data structures and preconditioners, as well as its own family of methods.

Figure 2 depicts the integrated software suite to be developed by the TOPS Center. The majority of its components already exist and several are very widely used; for example SuperLU and PETSc both have hundreds of active users. Others, such as the PDE-constrained optimization package VELTISTO, need to be "hardened" before they are widely used. A side-benefit of the TOPS center will be the far greater interoperability of the various components of the TOPS suite, as well as an integrated web site that leads application developers to the most appropriate combination of software components for their problem. But the main focus of the TOPS center will be far greater functionality and performance of the various solvers that we can provide.

2.7 Software Performance Tuning

No obstacle to progress in the world of scientific computing has proven more stubborn and frustrating than the sluggish pace of improvement in software performance. The underlying computing hardware doubles its speed every eighteen months, yet it often takes more than a year for software to be optimized or "tuned" for performance on a newly released CPU. Users tend to see only a fraction of the power available from any new processor until it is well on the way to obsolescence.

This problem was originally addressed by the PhiPAC project [21, 22, 134] for matrix-multiplication, the kernel of the Basic Linear Algebra Subroutines or BLAS [29, 68], the fastest and so most attractive kernel in high performance matrix computations. PhiPAC developed a system to create automatically-tuned matrix-multiply routines for RISC architectures, performing about as well or better than hand-optimized codes from vendors. This work held the promise of automatically optimizing many other routines on a variety of architectures, without the need for expensive and very slow hand optimization understood by a very few practitioners.

PhiPAC inspired the development of ATLAS (Automatically Tuned Linear Algebra Software). ATLAS [135] is a self-adapting package of numerical software that implements a new paradigm for software optimization. ATLAS replaces the months of arduous hand coding with special software that automatically rewrites itself in the space of a few hours to maximize its performance on a particular CPU. ATLAS has been so successful that it has been adopted by the Mathworks (producers of Matlab) and other vendors.

We have also addressed the important kernel of sparse-matrix-vector-multiplication, another central kernel in PDE computations. The SPARSITY system [90, 91, 92] takes as input a sparse-matrix and produces an optimized data structure tuned to that sparsity pattern, and a matching optimized multiply routine. Results show speedups of up to a factor of 5 over straightforward implementations.

3 Research Design and Methods

This section builds directly on the accomplishments of Section 2, in each of seven areas. We argue that users either expect a PDE solver package to offer all of the functionality proposed in this section, or they *should*. They will quickly exploit it if conveniently packaged. The co-PIs contributing heavily to a project are listed alphabetically in brackets at the head of each subsection. (Precise allocations of effort appear on institutional budget sheets.)

3.1 PDE Time Integrators

[Smith, Woodward] Adaptive time integration is an important aspect of multiphysics problems. Variable-order and variable-stepsize BDF methods have been shown to be highly effective for the stiff systems arising from such applications. Semi-discretization of time-dependent PDEs leads to a system of either ODEs or DAEs. General-purpose (sequential) software packages developed at LLNL are among the most widely used solvers for these problems.

Since adaptive time integration itself is otherwise supported at LLNL, we propose no additional algorithmic research as part of this project. However, the TOPS ETC will support (for PDE applications) the PVODE [45] and IDA [87] software for parallel solution of implicit ODEs and DAEs, respectively. PVODE has already been interfaced with the PETSc library, because of user demand. PVODE users will continue to have access to their favorite temporal control structures in a fully parallel solver environment.

3.2 PDE Nonlinear Solvers

[Cai, Keyes, Smith, Vassilevski, Widlund, Woodward] For nonlinear elliptic PDEs, many techniques are in use in applications groups at the national laboratories. The majority seem to be variations of block-iterative linear techniques, such as block Gauss-Seidel, utilizing basic nonlinear solvers, such as Picard iteration. Unfortunately, many important nonlinear problems, such as magnetohydrodynamics, porous media flow, and radiation diffusion, are not strongly elliptic, and they often contain non-elliptic components causing the solutions to be nonsmooth and to have local singularities, such as shocks and sharp fronts. Therefore block-iterative methods are not only slow, they may be altogether ineffective. The development of fast, scalable, parallel iterative methods and software for this class of highly nonlinear and mixed-type PDEs is a basic component of the TOPS Center.

We address these problems with three complementary algorithmic strategies: Schwarz preconditioned inexact Newton, inexact Newton-Krylov-multigrid, and full approximation scheme (FAS) multigrid. We envision that eventually most application codes will use a combination of the three approaches, and thus we will develop a common mathematical framework for describing all the methods (much as the abstract multigrid/Schwarz framework provides concise descriptions for many linear preconditioners). This mathematical system will be used to design and implement a general software system that allows users to easily switch at run time between the various approaches (much as the PETSc PC construct works for linear preconditioners).

3.2.1 Schwarz Preconditioned Inexact Newton

Experience with developing iterative methods for nonlinear problems shows that a major difficulty in solving nonlinear problems is often not in the nonlinearities, but in the singularities arising from the non-elliptic components. The singularities in the solution are often local, i.e., they appear only in a small part of the computational domain or involve only a small number of equations in a large nonlinear system. This observation leads us to a family of algorithms based on nonlinear elimination, in particular, nonlinear Schwarz-type domain decomposition methods, such as ASPIN [47]. Before applying Newton's algorithm, we nonlinearly precondition the global problem with local nonlinear solves (themselves solved with inexact Newton). The preconditioning step ameliorates the non-smoothness of the global problem. Preliminary tests have shown that this nonlinear Schwarz preconditioned inexact Newton algorithm is fast and robust for compressible and incompressible flow problems [47, 48] that do not converge from cold initial guesses using Newton's method in their original unpreconditioned form.

Though effective, in reducing the number of Newton steps, the single-level nonlinear Schwarz preconditioner does not reduce the condition number of the linear systems to be solved in a scalable way. Multilevel techniques are still necessary to make the approach useful for large-scale problems on massively parallel machines. Therefore, we will combine multilevel preconditioning together with Schwarz nonlinear preconditioning. The degree of accuracy at which the subproblems are solved also influences enormously the overall efficiency of the methods and is an area that will be investigated.

3.2.2 Newton-Krylov-Multigrid

Inexact Newton-Krylov techniques preconditioned with multigrid have been used successfully in a number of applications including porous media flow [94] and radiation diffusion [44]. For fast convergence, however, these methods require initial guesses fairly close to the solution. We propose to extend ideas from nonlinear multigrid to improve global convergence of the Newton methods. We will investigate the use of nonlinear coarse grid solves to provide effective initial guesses for the Newton techniques. Convergence of the resulting method will be analyzed, and if effective, software will be developed and deployed to application codes. This sort of grid continuation method will be compared with traditional globalization techniques for Newton's method, such as backtracking and trust region methods. We will also look at combining these strategies with other continuation methods.

Matrix-free finite-differencing-based Newton-Krylov methods require a nonlinear function evaluation for each linear iteration. If this function evaluation is expensive, the methods will be slow. We are currently investigating convergence properties of variants of these methods where some of the nonlinear coefficients are lagged in the function evaluations required in building the Krylov basis. We will build on this work to make these variants viable solvers for applications models.

When matrix-based Newton-Krylov methods are used, an unavoidable expense and difficulty is that of computing the Jacobian. Fortunately a variety of techniques are available to simplify the calculations. We will continue our work with automatic differentiation to generate solutions that do not require that the user provide explicit Jacobians. In addition, parallel colorings will be developed that allow the efficient computation of Jacobians via finite differencing.

3.2.3 Nonlinear Multigrid (FAS)

Nonlinear multigrid, or Full Approximation Scheme (FAS), methods [35] tend to have a larger radius of convergence than Newton methods, and they also offer the potential of avoiding the large expense of calculating the full Jacobian. For this reason, we propose to investigate the viability of nonlinear multigrid methods for systems of nonlinear elliptic and parabolic equations. Although these methods are not new, they have only recently come into use, as advances in the solution of heterogeneous problems have made them more applicable

to problems of interest.

Despite this recent success, many issues remain before these algorithms can be widely used. These issues include design of efficient interpolation and restriction operators, development of effective coarse grid operators, and efficient use of nonlinear smoothers. We will investigate these components of FAS methods with the goal of designing efficient algorithms with fast convergence. Tradeoffs concerning the reuse or re-computation of solution dependence in the interpolation and restriction operators, the use of point or coupled smoothers, and the re-discretization versus Galerkin coarse grid operators will be examined. For example, in linear multigrid methods Galerkin coarse grid operators give the optimal coarse grid update (in the energy norm). However, these coarse grid operators, in the nonlinear case, require evaluation of the nonlinear function on the finest grid. We will investigate alternatives to this type of coarse grid operator formulation and examine both theoretical and computational convergence of the methods.

The outcomes of these tradeoffs will likely depend on the interpolation and restriction operators, as well as the application. Our goal will be to determine general guidelines for choosing the right components of FAS for a given application. In this work, we will examine both structured and unstructured grid methods. For the unstructured case, we will investigate our AMGe technique (see §3.4.1) for nonlinear elliptic problems. This work will include the derivation of coarse finite element discretizations of nonlinear elliptic problems algebraically based on the element agglomeration technique developed previously for linear problems [93]. We will also explore various homogenization techniques that follow naturally from the AMGe algorithms proposed for linear problems.

Finally, we will investigate cycling strategies for these methods. Cycling strategies include the V-cycle, the W-cycle, and the full multigrid cycle. For linear problems, the full multigrid cycle can give rise to an optimally convergent method. For nonlinear problems, this strategy tends to widen the radius of convergence for the nonlinear solver, as the solutions to the coarse grid problems are more likely to be in the convergence region of the fine grid solution.

3.2.4 Common Framework for Three Complementary Approaches

The initial phase of the project will be on the design and software implementation of the proposed algorithms in a common abstract framework. This will allow combinations of the three approaches to be tried easily in application codes to find the most suitable. In the second phase, we will study three issues of importance for parallel iterative methods, namely their robustness, scalability, and competitiveness compared with existing methods. To study the robustness of the proposed algorithms, we will use a variety of test problems. Scalability will be demonstrated on thousands of processors. In order to document the competitiveness of the proposed algorithms, we will compare them with existing techniques for converging nonlinear problems that do not take advantage of hierarchical representations, such as parameter continuation, pseudo-transient continuation, and nonlinear robustification techniques, such as backtracking and trust region methods. General guidelines will be developed to allow application users to tune the solvers for their particular problems.

We will focus on three basic application areas. The first is fluid flows including compressible, incompressible and porous media. This area has been the driving force behind many advanced algorithm developments. The second area is radiation hydrodynamics. We will start with the radiation diffusion model [33, 117], a highly nonlinear phenomenon, involving sharp thermal fronts. Then we will couple the radiation diffusion with hydrodynamics. The third area is magnetohydrodynamics, in which the flow field is coupled with the magnetic field. Our partnerships with SciDAC applications groups (see Section 4) will ultimately dictate problem-specific concentration of effort in this polyalgorithmic domain.

3.3 PDE-constrained Optimization

[Benson, Biros, Cai, Ghattas, Keyes, Lee, Moré] Optimization of multi-component PDE-based simulations arises in data assimilation (e.g., climate modeling), optimal design (e.g., accelerators), parameter estimation

(e.g., reservoir models), and optimal control (e.g., fusion reactors) problems. This ETC will build on the TAO and VELTISTO toolkits, and develop techniques that integrate PDE and sensitivity analysis with optimization algorithms for high-performance architectures.

3.3.1 Optimization Methods

We will develop and extend our existing PDE-constrained optimization algorithms to inequality constraints and time-dependent PDEs. In addition, our algorithms will incorporate multilevel globalizations and tolerate approximate Jacobians and Hessians. Research issues associated with these tasks are outlined below. A major goal of this research will be CCA-compliance and interoperability of the TAO and VELTISTO toolkits.

A significant challenge for PDE-constrained optimization is the inclusion of (pointwise) inequality constraints that scale with the dimension of the state variable. An example is an optimal design problem with a constraint on the stress at each point in a solid. A difficulty is that conventional adjoint PDE-optimization algorithms require an additional PDE solve for each constraint that is active, thus destroying scalability when there are large numbers of active constraints. Interior point methods are a promising alternative but these methods tend to generate subproblems for which standard iterative algorithms and preconditioners fail, and thus we need to develop specialized preconditioners (see Section 3.4.3 for related work). An advantage of interior point methods is that the required number of (approximate) PDE solves is reduced to just two. Interior point methods for nonlinearly-constrained problems are under active development, but with few exceptions [110] have not been applied to optimal control problems. Other options include augmented Lagrangians and semi-smooth methods. We will compare these algorithms and extend them to PDE-constrained problems.

Another important thrust of our work is the use of hierarchical techniques in optimization algorithms. Traditionally, optimizers have used general-purpose line-search and trust region methods to assure global convergence. However, many PDE solvers use other techniques to assure robustness, including multiple grid sequencing, parameter continuation, and pseudo time-stepping. The use of such globalizations is likely to be more efficient than general-purpose techniques, and thus we will integrate them into our optimization algorithms. This research will be done in close collaboration with the work in Section 3.2.2.

We also will pay close attention to the use of approximate Jacobians and Hessians since, unlike simulation codes, optimization algorithms require the Jacobian of the constraints and its adjoint in the optimality conditions. Directional derivatives can be used whenever Jacobian-vector products are needed, but this does not work for adjoint methods, since the transpose of the Jacobian is needed. Moreover, optimal optimization solvers are likely to require the use of second-order information, and yet most constrained optimization codes use only first-order information. As part of this research we will develop parallel coloring algorithms for determining Hessian matrices in collaboration with the work on (non-symmetric) Jacobians in Section 3.2.2. Ultimately, the solution lies with automatic differentiation tools [75].

Another significant challenge in PDE-constrained optimization is handling time-dependent PDE constraints. The difficulty is that the optimality system includes a backward-in-time PDE system (the *adjoint* system) featuring source terms that stem from solution of the original forward-in-time state PDEs. Thus, the most work-efficient implementation requires storage of the entire time history of state variables, which is not feasible for large-scale problems. Checkpointing algorithms, based on ideas from automatic differentiation [74], are capable of trading off memory for work. They store carefully selected time snapshots in the forward sweep, and recompute intermediate vectors on the backward sweep [2]. The best checkpointing strategy to take in the context of other parallel PDE solver tradeoffs, including level of incomplete factorization fill, subdomain overlap, and depth of grid hierarchy, is not clear, and will be studied within this project.

3.3.2 Sensitivity Analysis

In addition to efficient methods for solving nonlinear, time-dependent PDEs, we will develop methods to assess the sensitivity of the PDE solution with respect to the given initial conditions and/or model parameters. This

is a desirable capability since it is common to have initial conditions and parameter values that are not known precisely. In particular, we will pursue a novel approach that builds on the wide variety of tools and techniques that are available for computing derivative or sensitivity information – e.g., automatic differentiation (AD) tools: ADIFOR, ADIC, TAMC; sensitivity solvers: SensPVODE, SensIDA, DASPK3.0; etc.

The main objective is to determine what types of initial perturbations can become greatly amplified during the time integration of nonlinear PDEs. In doing so, we can identify the sources of error (in initial conditions and parameters) that most greatly affect the PDE solution. In meteorological applications, these perturbations are sometimes referred to as the "leading singular vectors." To compute them, the basic approach requires methods for determining the sensitivity of the PDE solution with respect to initial conditions (forward sensitivities) and the sensitivity of the initial conditions with respect to the PDE solution (backward sensitivities). This process involves the use of forward and reverse mode AD tools, the development of tangent linear models and adjoint models, or other methods. The approach for computing the leading singular vector is based on the power method for computing eigenvalues and eigenvectors. The forward sensitivity of the PDE solution to an initial perturbation vector is computed, followed by the backward sensitivity of that result (with respect to initial conditions). In general terms, a single forward/backward sequence corresponds to multiplying the initial perturbation vector by a symmetric matrix. Additional iterations produce powers of the matrix times a vector and, if all goes well, the initial perturbation rapidly converges to the eigenvector corresponding to the dominant eigenvalue. This eigenvector is the leading singular vector that we seek; it indicates the type of perturbation that is likely to be greatly amplified in integrating the PDE.

3.4 PDE Linear Systems

In any PDE analysis, optimization, or eigensystem code are numerous linear systems to be solved (often only approximately). The DOE and other federal agencies have invested for years in the creation of software for linear systems. Such codes as HYPRE have paid off handsomely in reservoir and radiation transport applications. However, many important challenges remain, especially for multicomponent, multiphysics PDEs. This ETC will develop geometric and algebraic multigrid solvers, direct and approximate factorization solvers (including multilevel orderings for ILU), *a priori* pivoting strategies for nonsymmetric systems, and solvers exploiting the complex symmetric case (which arises in E&M and MHD applications).

3.4.1 Algebraic Multigrid Methods

[Falgout, Manteuffel, McCormick, Smith, Vassilevski, Widlund] The so-called algebraic multigrid (AMG) method [34, 36, 39, 40, 122, 123, 128, 129] assumes only information about the underlying matrix structure, and as such, is ideally suited for solving unstructured grid problems. There is no single AMG algorithm. In fact, AMG has come to describe a whole class of algorithms: those that use algebraic information in the matrix to construct the basic multigrid components. The AMG framework assumes the use of a simple pointwise relaxation method and attempts to correct the algebraically smooth error that remains after relaxation by suitable choice of coarse grids, intergrid transfer operators, and coarse-grid equations. To do this, AMG must be able to efficiently characterize this algebraically smooth error.

The "standard" AMG algorithm [123] uses heuristics based on M-matrix properties and works remarkably well for many important classes of problems, non-M-matrices included (see [55] for details). Unfortunately, there are important classes of problems for which this AMG algorithm performs poorly. To address this, the co-PIs have developed a new method called AMGe for solving finite element problems. This new method uses additional problem information, namely the local finite element stiffness matrices. With this extra information, AMGe can better characterize smooth error components, and this in turn allows for better coarse grid selection and better interpolation operator definition. While AMGe does use additional element information, it is otherwise more general and promises to be effective in cases where AMG and most any other MG scheme fails—particularly for systems with strong cross-variable coupling, a focus of this ETC.

The AMGe method uses a heuristic from multigrid theory that states that interpolation must be able to reproduce an eigenmode up to the same accuracy as the size of the associated eigenvalue. That is, an eigenmode with large associated eigenvalue need not be interpolated well, but an eigenmode with small associated eigenvalue does. Although this heuristic provides us with good guidance for constructing multigrid algorithms, it is not practical because it involves knowledge of the global spectrum of the operator. To address this, in AMGe we localize the heuristic so that we need only know about the spectrum of small local operators. The primary method for constructing these local operators is to sum together the finite element stiffness matrices in some local neighborhood. See [41] for more in-depth information about the basic AMGe method, theoretical results, and an efficient method for constructing interpolation in practice. See [93] for details on an element agglomeration algorithm for constructing the coarse elements and stiffness matrices needed to do multilevel AMGe.

We propose to develop new AMGe and related methods. The goal is to combine this work with our expertise on parallel AMG software in HYPRE to enable the robust, scalable, parallel solution of large-scale SciDAC application problems. Examples of problems that are particularly difficult for multigrid methods include Helmholtz, which can have algebraically smooth error components that are geometrically very oscillatory, and problems with anisotropies that are not grid-aligned. A new method that we have only begun to investigate is called spectral AMGe. The spectral method differs from the previous versions of AMGe in that the degrees of freedom are not nodal (i.e., not associated with a subset of the fine degrees of freedom); rather they are coefficients in the expansion of algebraically smooth vectors in terms of a coarse smooth basis. The coarse basis is constructed locally based on eigenvectors of local (small) matrices, and requires an agglomeration procedure similar to the one mentioned above. Preliminary tests indicate that the new method with standard smoothing (like Gauss-Seidel) is very efficient and extremely robust. Particularly promising is the fact that the new spectral method's performance seems to be insensitive to the way the agglomerates are constructed.

One special version of spectral AMGe is the spectral agglomerate AMGe method. In this method the concept of elements and element matrices is preserved at all coarse levels. This is achieved by restricting the interpolation rule to be compatible with the topology of the agglomerated elements (the shared degrees of freedom by two or more agglomerates are uniquely interpolated). Initial tests on highly anisotropic elliptic problems show promisingly-high robustness of the method. The method offers the option to explicitly build a hierarchical basis (HB) that may in turn be used to do more sophisticated smoothing based on ILU or sparse approximate inverses. These smoothers smooth the error components in a hierarchical complement to the coarse space, and the HB version of the method seems to be more robust than the original version (albeit more expensive). However, it may be needed for more general (not inherently elliptic) systems of PDEs, like the ones obtained by FOSLS discretizations of non-elliptic dominated problems (e.g., Helmholtz, convection-dominated convection-diffusion problems as well as nearly-incompressible elasticity). One also notices that a simple version of the spectral agglomerate AMGe rule can provide a good set of tentative interpolants needed in the smoothed aggregation method [132], called spectral smoothed aggregation MG.

The AMGe methods mentioned above will be tested and compared on a broad range of application problems, including problems obtained by the FOSLS discretization technique. The central trouble with many applications is reduced ellipticity. In applications like particle transport, highly convective flow, and electromagnetics, approximate null spaces of the discrete operator do not exhibit the smoothness expected in elliptic systems. AMGe should be able to determine the local character of these near null space components to obtain a fast solver for the discrete equations. One of the central issues is whether we can develop an AMGe scheme that can apply to nonsymmetric problems and saddle–point problems coming from Stokes and mixed finite element methods. A compelling alternative is to apply FOSLS to the problem to create that symmetry, as well as positive definiteness. The combination of AMGe and FOSLS could be a powerful tool for applications where FOSLS has been unable to obtain full H^1 ellipticity.

Because the AMGe methods require additional local stiffness matrix information, traditional "linear-algebraic" solver interfaces are not sufficient for providing these methods in a library setting. However, new

solver interfaces such as the Finite Element Interface (FEI) [53] are ideal for packaging these AMGe algorithms for the Office of Science user community.

Another class of methods to be investigated is based on the concept of compatible relaxation, recently proposed by Achi Brandt [37]. In its simplest form, this is just F-relaxation, i.e., relaxation at those fine-grid points that are not also coarse-grid points. The basic idea is as follows. If compatible relaxation is fast to converge, then the coarse grid is adequate for eliminating the remaining error. If it is not fast to converge, then either additional smoothing (or more aggressive smoothing) can be done, or more points can be added to the coarse grid. It turns out that fast convergence of F-relaxation is equivalent to satisfying the global AMGe heuristic mentioned above. We will consider new algorithms that utilize this idea in the hopes of finding a robust efficient method that requires little or no additional information aside from the global system matrix.

Related to AMG methods is the balancing Neumann-Neumann method [103] for unstructured grids. We have developed a comprehensive set of mathematical tools for analyzing this method and recently incorporated a robust, highly parallel implementation for scalar PDEs into PETSc. We will complete work in this area by analyzing and implementing these preconditioners for multi-component PDEs, for example, for the Navier-Stokes equations. This method will be compared with the classic AMG and AMGe methods.

3.4.2 Geometric Multigrid Methods

[Falgout, McCormick] For problems that are structured or mostly structured, geometric methods have the advantage of producing efficient computational kernels and cheaper storage requirements than for more general methods such as AMG and AMGe. The disadvantage with geometric methods is that structure must be maintained on all grid levels, restricting the possibilities for coarse-grid selection and making it difficult to handle problems with unstructured physics such as problems with multiple directions of anisotropy. We propose to develop geometric algorithms in support of structured and semi-structured application codes in the Office of Science, including block-structured codes, structured adaptive mesh refinement (SAMR) codes such as those being considered in the AMR ETC, and codes using the overset grid technologies being considered in the TSTT ETC.

Applications using SAMR have traditionally had difficulties getting good linear solver library support, and have typically had to develop home-grown implementations of algorithms such as the Fast Adaptive Composite grid method (FAC) [107]. Two semi-coarsening multigrid methods were developed earlier by the co-PIs [8, 42] and have been used by several SAMR customers via HYPRE to do so-called level solves in FAC. But, improved efficiency could be had by providing the entire FAC solver within the linear solver library. The main difficulty is developing an interface between the SAMR application and the solver library that is both natural for the application user and that provides the necessary structure and hierarchy information needed by the solver to apply geometric methods.

The co-PIs developed and implemented (in parallel) a draft of such a semi-structured grid interface, but the focus to date has been primarily on block-structured grid applications. We propose to extend this interface to support SAMR and overset-grid applications, and to collaborate closely in this endeavor with the Applied PDE and TSTT ETCs and the ESI Forum. An added advantage of such an interface is that non-geometric linear solvers like ILU or AMG can also be provided "under the hood" without requiring that the application code be rewritten.

Once the problem of interfacing to these semi-structured applications is solved, there is still much work to be done developing solvers (like FAC) that take advantage of the structure present and that are robust and efficient. We propose to investigate a number of ideas, including domain decomposition, but one approach we intend to pursue in particular is a hybrid geometric-algebraic multigrid approach. We first consider applying algebraic multigrid to the entire domain. In regions where structure is present, algebraic multigrid usually results in coarsenings that are also structured. In the hybrid method, we propose to do this explicitly, and exploit the structure imposed upon the coarse grid. This method will produce coarse level problems with a

splitting into structured and unstructured components similar to that present in the original, fine level problem. In defining the components of this hybrid geometric-algebraic multigrid method such as interpolation and coarse grid operators, we build on our research in both structured and unstructured multigrid solvers.

We also intend to pursue new variants of the FAC method. This method can be thought of as a multigrid solver for a global problem discretized uniformly at the resolution of the finest patch; however, computation away from the refinement region is suppressed. Because of this close connection, our previous geometric multigrid research is relevant in designing scalable FAC solvers that are robust with respect to anisotropic and discontinuous coefficients.

3.4.3 ILU Preconditioners

[Chow, Demmel, Li, Ng, Pothen, Vassilevski] Incomplete LU (ILU) factorization preconditioners are sometimes more robust than other preconditioners on general problems, and are the preconditioners of choice in many applications. Block-ILU methods that incorporate coarse-grids (in the Schur complement computation throughout the factorization process) were proposed in the past (cf. [51]) and tested on some model problems, such as 2-D elasticity [50] and FOSLS discretization of convection-diffusion problems [49]. These results showed that building a coarse hierarchy, by algebraic means in general, leads to improved convergence properties of the resulting block-ILU factorization methods. Recently, other researchers have shown that ILU preconditioners exploiting multilevel hierarchy in the ordering of the unknowns exhibit MG-like convergence rates on model problems [10, 16, 17, 31, 32, 121]. These results have spurred much interest in this area, with the objective of devising a method that would combine the scalability of multigrid with the improved robustness of the ILU methods. It is clear that these methods have much in common with AMG but in general assume less knowledge about the origin of the algebraic problem and target PDEs that are not necessarily elliptic. Such techniques may be the key for large-scale multicomponent simulations such as fusion and combustion, where standard multigrid methods have not been effective.

Whereas earlier studies used multilevel reorderings of ILU for model problems, complex applications require more sophisticated ingredients. We propose transforming the original matrix problem into a generalized hierarchical basis (HB) [18], where the problem is much better conditioned. An ILU factorization is then carried out on this transformed system. Very little additional cost is required in both constructing the preconditioner and recovering the preconditioning operation in the original, nodal basis. Unlike multigrid, simple methods can be used for coarsening and interpolation, the reason being that the ILU process corrects any errors made in the transformation stage. (In fact, the preconditioner can be made exact no matter how poor the transformation is.) A perfect hierarchical basis giving optimal convergence rate may be impossible to construct for many unstructured problems; an imperfect basis coupled with ILU may lead to reliable preconditioners giving non-optimal, but practical convergence rates. Related ideas have been investigated by Bank and co-authors [16].

A second approach to taking advantage of the improved robustness of ILU in multilevel preconditioners is to use them as local solvers or smoothers in a multigrid framework. Block-matrix factorization [9, 56], which was developed in the mid-1980's, can be reformulated as multigrid by adding smoothing steps and appropriately choosing the block partitioning. Recently, we have shown that both ILU and sparse approximate inverses under certain conditions can be used in these block-matrix factorizations and give optimal convergence rates on model problems [52]. (Sparse approximate inverses were chosen because they make the preconditioner easier to parallelize, and automatically give sparse approximate Schur complements.) For more complex problems, we propose investigating the need to increase the accuracy or change the dropping criterion on coarser and coarser grids.

For challenging problems, it may also be necessary to improve the accuracy of the sparse approximate inverses and ILU factors while keeping them very sparse. In this context, it is necessary to study reordering schemes that consider not just the structural, but the numerical aspects of the problem. One of the few ex-

amples of this is the greedy algorithms in [58] which use properties of Schur complement matrices to guide the ordering process. Although accurate ILU factorizations giving improved convergence behavior could be computed, these algorithms were very costly to run. Building on our earlier work on efficient approximations to greedy ordering heuristics for full factorizations [112] we plan to incorporate numerical information into these approximations and test their effectiveness for ILU factorizations.

In addition to the above, we will continue our work on parallel implementations of memory-efficient ILU factorizations for symmetric matrices and domain-decomposition approaches for nonsymmetric matrices [89]. We also plan to add incomplete factorization functionality in both serial and parallel SuperLU. This work will be performed with the purpose of integrating high-performance ILU codes into multilevel frameworks.

Many preconditioning techniques are still problem-specific and of trial-and-error nature. We plan to target our algorithms for the specific applications in this proposal, such as accelerator design. An important part of the project is to also compare our approaches with others that are available. We will integrate the best codes into the PETSc and HYPRE packages and make them available to application codes.

3.4.4 Sparse Direct Methods

[Demmel, Li, Ng, Pothen] In a multilevel or multigrid method, sparse direct methods are reliable alternatives for solving coarse grid problems. In addition, sparse direct methods are extremely powerful and robust, particularly for matrices of moderate to large dimensions and for matrices are ill-conditioned.

We plan to incorporate robust pivoting strategies in both the left-looking solver (PANFCT) and the multifrontal solver DSCPACK so that they can handle general symmetric indefinite matrices.

We have studied the problem of optimizing memory resources required by sparse matrix algorithms. We have developed and analyzed efficient external memory sparse matrix algorithms. The results have led to the development of I/O efficient algorithms for sparse direct solvers [66]. We will extend OBLIO, our object-oriented sparse solver library with implementations of these algorithms for terascale computers. We will make use of OpenMP and MPI to make our implementations portable.

The current 2D matrix-process layout in SuperLU/GESP is critical for maintaining high efficiency on terascale computers. On the other hand, the 2D sparse matrix distribution requires nontrivial overhead in bookkeeping, which makes the solver not so efficient on small numbers of processors (e.g., smaller than 32). We plan to design a version of the solver with 1D data distribution. Intelligent decisions will be made at a higher level to choose between 1D or 2D, depending on the architectural features and solver needs (such as for coarse grid solution or preconditioning in a subdomain).

While the numerical factorization in SuperLU/GESP is completely parallelized, the symbolic factorization and analysis is not. On the very large machines targeted in this proposal this can become a bottleneck because of time or more importantly space, since no single processor may have enough memory to perform the symbolic factorization. In other words, parallelization with even quite modest speedup becomes necessary for memory scalability. We will work on this, noting that approximations to the matching algorithms used are often good enough, so that the parallel and sequential semantics may not exactly match.

Selective inversion [84, 119], which is a special form of partitioned inversion [3], has been employed successfully to resolve the latency bottleneck caused by multiple solution of sparse triangular linear systems that have the same coefficient matrix. The idea is to replace the substitution process by matrix-vector multiplications, and hence make parallel triangular solutions more scalable. The current triangular solution phase in SuperLU/GESP uses the same 2D matrix distribution as in the factorization phase. The more efficient scheme is 1D distribution in this phase so that partitioned inversion or selective inversion can be incorporated relatively easily. It may be worth adding a data re-distribution phase between factorization and triangular solution. The overhead with data re-distribution or selective inversions can be well amortized if more than one solution is required. Partitioned inversion or selective inversion can also be incorporated into the triangular solution phase of PANFCT to improve efficiency.

For GESP, static pivoting is achieved by finding an *a priori* numerical ordering. Sparsity is maintained by choosing a symmetric ordering using structural information. We plan to develop symmetric ordering algorithms (that preserve the main diagonal) that can be computed from the structure of A alone. The algorithms are based on local greedy heuristics analogous to Markowitz's scheme, but the nonsymmetric structure will be maintained. In a later stage, we will also investigate graph partitioning heuristics for this ordering purpose.

In the case of sparse symmetric matrices, we plan to implement a 2-step parallel ordering strategy. The first step will be based on parallel versions of nested dissection schemes [80, 83, 95, 118] to generate domains, while the second step will be based on greedy methods [4, 102, 112] applied to the domains. Such hybrids can be quite effective at reducing fill [85]. The impact due to the choice of separators will be studied. Strategies for choosing separators, including the notion of look-ahead, will be considered.

3.5 PDE Eigenanalysis

[Demmel, Husbands, Marques, Ng] The problem of finding a number of selected eigenvalues and corresponding eigenfunctions of a differential operator arises in many applications [62, 105, 133, 100, 136]. One particular area that is critical to DOE's mission is large-scale computational modeling of the electromagnetic field within a particle accelerator. Eigenanalysis also comes up in fusion energy sciences and chemical sciences.

We plan to work on the generalized eigenvalue problem $Kx = \lambda Mx$, where the mass matrix M is sparse and often symmetric and positive semi-definite. For the accelerator physics simulation, the stiffness matrix K is complex symmetric.

Typically, a small number of eigenpairs are desired. A projection method designed to extract desired spectral information from a small-dimensional subspace is appropriate. The shift-invert Lanczos or Arnoldi iteration combined with implicit restart (as implemented in ARPACK) is generally effective for small to medium sized problems. For truly large problems, the limiting factors are usually the memory requirement for decomposing the sparse matrix $K - \sigma M$ into a product of triangular factors and the efficiency of forward and backward substitutions. The recent effort [140] in combining DSCPACK and PARPACK to calculate the vibrational modes of large-scale molecular systems on distributed memory machines has indicated that reasonable parallel speedup can be achieved by incorporating selective inversion [119] into sparse Cholesky factorization. Our near-term goal is to integrate sparse direct solvers to be developed in this project into sparse eigenvalue solvers so that a large variety of generalized eigenvalue problems can be solved efficiently using shift-invert Arnoldi iterations.

When the size of the problem reaches a level for which shift-invert becomes prohibitively expensive, it is natural to consider replacing the direct solver with a preconditioned iterative solver. However, there are difficulties associated with this approach:

- One must identify an effective preconditioner to solve a linear system of the form $(K - \sigma M)w = v$.
- Even with a good preconditioner, one must solve the linear system to a high accuracy to maintain all the desired properties of a Krylov subspace (even though the accuracy required in the approximate eigenpairs is well below machine precision). This could potentially be very expensive.

The recently developed Jacobi-Davidson (JD) [125] algorithm addresses the second issue to some extent. The algorithm can be described as an inner-outer iteration. Approximations to the desired eigenvalues and eigenvectors are computed in the outer iteration by projecting the matrix pencil (K, M) to a subspace V. The basis vectors of the subspace are constructed by a sequence of inner iterations that provide approximate solutions to a linear system of the form

$$(I - (Mq)(Mq)^T)(K - \theta M)(I - (Mq)(Mq)^T)z = -r, \;\; q^T Mz = 0, \tag{1}$$

where (θ, q) is the current approximation to a desired eigenpair and r is the current residual, i.e., $r = Kq - \theta Mq$. The solution (z) to these equations can be viewed as a first-order correction to q. Hence the JD algorithm shares some similarity with Newton's method. The algorithm has been used successfully in a number of applications [30, 131] including the recent work in computing normal modes of large-scale molecular systems [139]. It has been observed that one does not need to solve the correction equation (1) to full accuracy in order to produce a reasonable subspace for eigenvalue approximation. However, the issue of how to control the accuracy of the inner iteration and thus balance the work between the inner and outer iterations remains an open question that we would like to address in this proposed research.

Another approach to interior eigenvalue calculation is related to the recent developments in Truncated RQ (TRQ) [127] and Truncated QZ iterations (TRZ) [126]. We plan to make further improvements by extending the basic TRQ and TQZ algorithms to a block variant that will allow us to capture clustered eigenvalues more efficiently.

The current generation of eigenvalue calculation software typically aims at solving a matrix eigenvalue problem through algebraic manipulations. These software packages tend to ignore properties of the underlying PDE from which the matrix eigenvalue problems are derived. A major component of this proposed work is to develop algorithms and software that will allow us to explore properties of the PDE in a multilevel framework. (This is especially appropriate for the accelerator simulation in which the differential operation associated with the electric field to be modeled is elliptic in nature.)

There has been some effort in devising effective multilevel methods for large-scale eigenvalue calculations since the late 70's [81, 15, 38, 104, 108]. Some recent work [138] has been done on combining a multigrid solver with the Jacobi-Davidson algorithm to compute the interior eigenvalues of an elliptic differential operator. A multigrid solver can be viewed as a preconditioner for the inner iteration. Because the correct equation (1) has a special structure, the construction of the interpolation and restriction operator must take the additional projection $I - (Mq)(Mq)^T$ into account. An interpolation and restriction operator suitable for solving a projected equation has been developed in [138]. One of the main advantages of using a multigrid solver for the inner iteration is that it can be parallelized easily on distributed-memory machines. Scalable performance can be achieved on a variety of platforms with hundreds or thousands of processors. In addition to using multigrid in the inner iteration, we also plan to investigate the convergence property of using a standard iterative solver combined with a multilevel ILU preconditioner in the inner iteration.

Like Newton's method, the correction process in JD is a local procedure that converges rapidly only when the starting point is sufficiently close to the solution. The global convergence of the JD algorithm assumes that approximate eigenpairs are available. Under a multilevel framework, the initial approximation can be obtained by solving the eigenvalue problem on a coarse grid first (using perhaps shift-invert Arnoldi) and interpolating the coarse-grid solution to a fine grid.

This approach is already used in the SPAM (Subspace Projected Approximate Matrix Modification of the Davidson Method) [124], developed at ANL. The method is based on subspace projections of a sequence of one or more approximating matrices. These matrices represent different levels of approximation of the exact matrix. The role of the approximating matrices is to improve the efficiency of the solution of the desired eigenpairs by reducing the number of matrix-vector products that must be computed with the exact matrix. Thus, even though more matrix-vector products may be required with this approach, the number of exact matrix-vector products and the total computation time can be reduced. The SPAM method is related to multigrid-like techniques and cascadic methods.

We propose to develop a parallel implementation of SPAM and exploit preconditioning technology within the PETSc software environment. We also will investigate the extension of this approach to the solution of linear systems. This work is collaborative with R. Shepard and A. Wagner (DOE/BES). Application of the work proposed here specifically focused on computational chemistry problems dealing with MRSDCI calculations and CRP simulation will be conducted under the "Advanced Software for the Calculation of Thermochemistry, Kinetics, and Dynamics," (A. Wagner, Principal Investigator) SciDAC center.

3.6 PDE Software Integration and Deployment

[Falgout, Ng, Smith] Under the TOPS ETC, the software will be developed, maintained, and deployed in a hierarchical manner. The framework will encompass three software suites (one for LBNL, ANL, and LLNL) and each suite will contain several packages. A single unified website will allow users to quickly obtain their required software, documentation, and contact information. Each package will have its own support team with a direct e-mail route for contact. If the author of a package, for whatever reason, is no longer able to support the package, responsibility for its maintenance will be transfered to the team of another package in the same laboratory suite. If a laboratory, for whatever reason, declines to support any of its packages, the responsibility, along with associated funding, will be transfered to an alternative. Additional obligations with regard to the TOPS software are listed in Section 4.2. (It is implicitly assumed here that universities are generally not set up to maintain software over the long-term of transient personnel and project funding. Hence, while innovation from the universities is critical in all other areas of this project, activities closely related to code integration, maintenance, and support of university-originated codes will be taken over by the laboratory with the closest associated code suite.)

Figure 2 lists against axes of laboratory (across) and algorithmic family (down) the software suites we propose, as of this point, for development and maintenance. *Some* of the interoperability that we already have or anticipate is shown. (The arrows in this figure show the main interoperability capabilities of the PETSc Suite, as an example. There are many other interoperability combinations planned involving the other suites that are not shown here. Interoperability with the products of other ETCs and applications codes is not shown at all.)

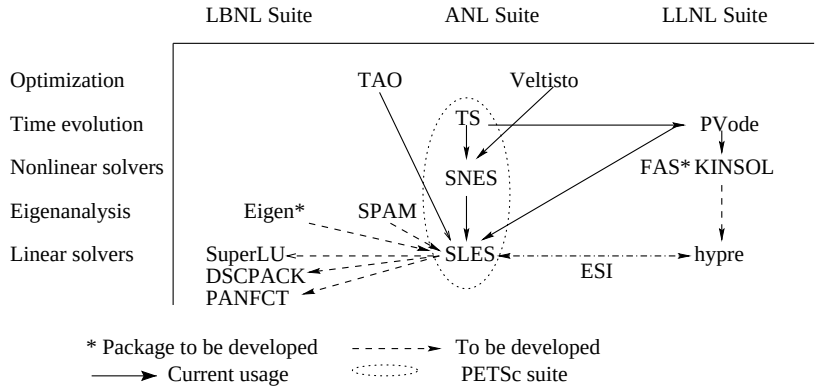

Figure 2: *An arrow from A to B indicates that A uses B. Asterisked packages and dashed interoperability connections are to be developed as part of TOPS. The other packages exist in at least stand-alone form, but will be further developed, integrated, and supported as part of this project.*

All new software developed will use MPI (and when appropriate, OpenMP) to provide the scalable parallelism. Most of the software will be written in C or C++ though portions may be written in Fortran 77/90. Each package group will develop and support interfaces for C, C++, and Fortran. In most cases these will be relatively simple wrappers that map between languages. The interfaces may also be provided via a SIDL (scientific interface definition language) implementation: the LLNL package BABEL. All software will conform to the CCA, ESI, and FEI standards, allowing easy interoperability with other compliant software. It is important to note that conformity with CCA, ESI, and FEI implies existence of a set of interfaces for the package that match the appropriate standard. A package may also provide other custom interfaces. In general, we expect each package to provide several interfaces for different classes of users.

To ensure code portability, correctness, and performance, we will develop an automated test system as we

develop the interfaces and implementations; to ensure conformance to the ESI and FEI standards, we will use their corresponding test suites.

In order to propagate the usage of the TOPS software we will provide two "integrated" tutorials via the Access Grid each year. These tutorials will *not* be organized around packages (wherein each author gives a separate presentation of his or her package) but rather around *computational tasks*. For example, one presentation may be on solving sparse blocked linear systems and another on optimization subject to a PDE constraint. The materials (slides and demos) for the tutorials will be available and maintained at the TOPS website.

3.7 PDE Software Performance

[Demmel, Dongarra, Keyes, Smith] Straightforwardly implemented multilevel methods are generally memory bandwidth limited and thus seriously underutilize a CPU's floating-point system (getting perhaps 10 percent or less of the peak floating-point performance). This ETC will investigate novel algorithmic variants and implementation techniques to take advantage of the cache-based memory hierarchies in today's systems. Our experience hand-tuning grid-based codes [77] will feed into our automated matrix-based approaches. We will work closely with the Performance and Benchmarking ETC to standardize and propagate optimization techniques. The ATLAS system will be extended to provide automatic techniques for producing optimized code for various processors.

3.7.1 Automated Empirical Optimization of Software (AEOS)

Our proposed research on a method for the Automated Empirical Optimization of Software (AEOS) addresses this problem directly by providing the knowledge and tools necessary for creating software performance optimizations that are both portable (across differing architectures) and persistent (across evolving versions of the same architecture). The AEOS method replaces difficult and tedious hand-coded tuning with an approach that combines automated code generation and experimental timings to generate a systematic, high definition mapping of the key parameters of a target hardware platform, and then uses the map to optimize the performance of selected kernels and software operations. Having already developed a highly successful proof-of-concept prototype for linear algebra, our goal in the research proposed here is to expand the limits of performance portability using the AEOS method, laying the foundation for a new generation of self-adapting high performance libraries and contributing to the development of innovative compiler strategies for scientific computing. In particular we will:

- Conduct fundamental research that extends and perfects the AEOS method by enlarging the range and power of its automated optimization techniques, including the addition of capabilities for application-specific, run-time, and parallel tunings;
- Encode the positive results of this research in software, both by integrating them into our model AEOS library (ATLAS) for linear algebra, which is already in wide distribution, and by generalizing them to create an AEOS Toolbox that programmers can use to make software components of all kinds self-adapt for high performance; and
- Widely disseminate both our discoveries regarding the AEOS method and the software that encodes them so as to dramatically narrow the gap between the pace of processor development and application speed up for a broad range of scientific software.

The proposed research will push the exploration of the AEOS methodology forward along four complementary lines. First, it will expand the capabilities of our ATLAS prototype to cover more operations (e.g. packed, banded, and Level 1 BLAS) and to permit application-specific customizations. This work will both lay the necessary foundation for the other two phases of the effort and also provide the scientific computing

community with the immediate benefits of new AEOS-optimized libraries for these important cases. Second, it will support an in-depth investigation of the potential of run-time extensions of the AEOS methodology, focusing on the difficult but critically important area of sparse matrix computations. Third, it will begin to extend the AEOS method from serial to parallel computations, examining the possible benefits to be derived from the optimization of communication libraries. Finally taking the next logical step, we will explore a range of numerical algorithms for exploitation of the AEOS methodology, especially multilevel methods for multi-component systems of PDEs. While we believe that the progressive expansion and generalization of the AEOS method across these four areas of interest will naturally lead to additional optimized libraries for the community, an important, overarching goal of the work is to clarify and formalize the different aspects of the AEOS method so as to facilitate the incorporation of its results into the ongoing work of the hierarchical algorithm software and library development communities.

3.7.2 Problem-dependent Tuning

An important design issue in tuning is the extent to which the tuned algorithm depends on problem details known only at runtime. In the case of matrix-multiplication, the problem dimensions are only known at runtime, but tuning can still be done off-line. In the case of sparse-matrix-vector-multiplication [90, 91, 92], having the sparsity structure is essential, and it is probably only known at runtime.

Higher-level algorithms can also be tuned based on runtime data. Many ILU algorithms take as input several parameters that determine which non-zero elements to keep in the factors. Since choosing the best settings for these parameters is a complex process that frequently involves trial and error, we believe a system that could examine the performance results over a range of parameter settings and then suggest good parameter values would be valuable.

3.8 Milestones

Milestones are given in some detail for Years 1 and 2 and in less detail for Years 3 through 5. Milestones for code packaging, maintenance, support, and training are omitted from this year-by-year list since they are implicit; see §4.2. The granularity of "milestone" here is approximately one per university, and two to three per lab, per year.

- Year 1:
  - Develop a common mathematical framework for describing the FAS, truncated Newton-Krylov-Multigrid, nonlinear Schwarz families of algorithms.
  - Complete a prototype algebraic nonlinear multigrid code leveraging current linear algebraic multigrid software.
  - Document a convergence comparison of Newton-Krylov, nonlinear multigrid, and Picard nonlinear solvers for nonlinear elliptic or parabolic systems arising in an application of interest.
  - Design, implement, and test on thousands of processors a common interface for ADIC, TAO and VELTISTO libraries to enable steady-state PDE-constrained optimization.
  - Benchmark the use of interior-point and augmented Lagrangian methods of selected PDE-constrained problems.
  - Document the status of research on new compatible relaxation AMG algorithms, assessing the potential of this approach for solving difficult problems such as Helmholtz and systems of PDEs.
  - Demonstrate the use of parallel FAC and AMG algorithms on structured AMR problems using the semi-structured grid interface.

- Document the performance of spectral AMGe on systems of elliptic PDEs.
- Complete parallel implementations of ILU factorizations.
- Incorporate 1D data distribution into SuperLU/GESP.
- Improve the performance of parallel triangular solution in PANFCT, SuperLU, and GESP using partitioned inversion/selective inversion.
- Incorporate state-of-the-art sparse direct solvers into shift-invert Arnoldi codes, such as ARPACK and PARPACK.
- Evaluate and benchmark the new eigensolvers using an accelerator physics application and others.
- Implement a prototype HB ILU code; use this code to test the usefulness of the algorithm on various problems. If appropriate, determine strategies for its parallelization.
- Complete algorithmic design of Holy Grail eigensolver.
- Document automatic performance tuning studies on computational kernels of most interest to partner SciDAC users.

- Year 2:
  - Complete a new version of KINSOL providing for application of grid continuation in the context of nonlinear elliptic problems and for lagging nonlinearities in the Krylov basis.
  - Document performance and convergence comparisons of grid continuation methods applied to Newton's method and of lagged nonlinearities in the Krylov basis for Newton-Krylov methods with standard Newton's method and Picard iteration.
  - Design and prototype parallel interior-point methods for inequality constraints in PDE optimization.
  - Evaluate the use of multigrid techniques for optimization problems with PDE-constraints.
  - Design and implement optimal checkpointing schemes for adjoint time-dependent PDE sensitivities and optimization.
  - Extend SensPVODE to compute adjoint sensitivities for several time-dependent, application-based test problems.
  - Document the parallel performance of FAC solvers in structured AMR SciDAC codes using the semi-structured grid interface.
  - Extend the spectral AMGe method to FOSLS formulations of non-elliptic dominated PDEs.
  - Benchmark ILU preconditioners against other work and incorporate the best preconditioners into HYPRE and PETSc.
  - Implement a multilevel block ILU code, including different dropping strategies at each level, and different local reorderings to improve accuracy.
  - Incorporate parallel symbolic factorization into SuperLU; produce ILU version of SuperLU.
  - Implement symmetric Markowitz's scheme for nonsymmetric matrices.
  - Benchmark the 2-step ordering algorithm that combines parallel nested dissection and local greedy heuristics.
  - Incorporate multilevel ILU preconditioners into our eigensolvers, and evaluate and benchmark the eigensolvers.
  - Provide tuned kernels for particular DOE platforms.

- Years 3–5:
  - Add optimizations to integrated software tools for nonlinear systems using AEOS infrastructure.
  - Incorporate nonlinear additive Schwarz into the LNKS optimization code, as a nonlinear preconditioning, and compare against more conventional robustification techniques for constrained optimization.
  - Complete an initial parallel version of a structured grid nonlinear multigrid code leveraging linear structured grid multigrid software technology.
  - Document convergence comparisons of Full Multigrid cycles, V-cycles and W-cycles for nonlinear multigrid methods on a suite of application test problems.
  - Release a parallel, algebraic nonlinear multigrid code with options for fully coupled and point smoothers and solution-dependent interpolations targeting nonlinear elliptic systems with scalar and tensor nonlinearities.
  - Implement, harden, deploy, and test very large scale optimization algorithms for time-dependent PDEs with inequality constraints
  - Incorporate and benchmark multigrid techniques in combination with interior-point methods for PDE-constrained problems.
  - Complete an extension of SensPVODE to successively apply forward and adjoint sensitivity techniques for identifying the most greatly amplified initial errors within a simulation.
  - Document a performance comparison of using sensitivities within a power method and a Lanczos iteration for estimating maximal error amplification for PDEs.
  - Extend AMG and AMGe work to structured AMR and overset grid problems to develop robust new methods that exploit grid structure for added efficiency.
  - Extend AMGe to saddle-point problems coming from mixed finite-element discretizations.
  - Implement multilevel approximate inverse preconditioners.
  - Implement ordering algorithms for ILU and approximate inverses, and evaluate their effectiveness on problems from fusion applications.
  - Investigate graph partitioning ordering heuristics for nonsymmetric matrices in the context of GESP.
  - Develop criteria for controlling the accuracy of the inner iteration in Jacobi-Davidson algorithm and evaluate the criteria using the accelerator physics application.
  - Develop and implement block variants of TRQ and TQZ algorithms, and evaluate their effectiveness.
  - Incorporate Holy Grail into sparse eigensolves.
  - Generalize performance tuning to other kernels on arbitrary platforms.
  - Provide feedback to our SciDAC ETC partners on their software and respond to their feedback on ours, so that "the rails meet and the golden spike is driven," and the new functionality being created throughout the SciDAC initiative can be combined.
  - Develop tutorial examples from experiences with our SciDAC application partners that demonstrate how to apply our software and that illustrate some of the valuable insights available from sensitivity, optimization, and modal analysis studies.

Our ultimate milestone is restated from Section 1: *We intend through these steps to upgrade the performance and practical usability not just of PDE solvers, but of the scientific process of which numerically simulating a PDE is a part.*

4 Consortium Arrangements

4.1 Other SciDAC ETCs

We have special interests in the interaction of multilevel solvers with adaptive mesh refinement (AMR) technology, since both exploit a mesh hierarchy, for complementary reasons. We have exchanged letters (see Appendices) of commitment with the AMR ETC, which plans to dedicate personnel to interact with our ETC on this issue. We will also work closely with the TSTT ETC developing the most appropriate abstractions and code to manage "mesh hierarchies" needed by multilevel methods.

The tools developed by the Performance and Benchmarking ETC will be applied to and directly influence our work on high performance for multilevel methods. We share a co-PI with this ETC.

The tools of the Visualization ETC will enable runtime and post-processed advanced visualization of the PDE solutions. Moreover, parallel runtime visualization is critical to understanding and improving the performance of iterative solution methods.

Tying this all together will be the Common Component Architecture (CCA) ETC and the two standards for solvers: ESI and FEI. As discussed above, all of our software will provide interoperability via these DOE standards.

4.2 SciDAC Application Groups

We plan formal ties to three applications efforts under SciDAC, and have exchanged letters of commitment with them (see Appendices). The Princeton fusion group has been using PETSc for several years and has committed to work closely with the ETC as we expand the software base. Co-PIs of the LANL fusion group work already with the TOPS ETC Director [99, 109] on physics-based matrix-free Newton-Krylov solvers, and future collaborations between these groups will be coordinated for complementarity. The ORNL-based astrophysics SciDAC team will work with the TOPS Center to get needed algebraic linear solvers into their simulations. The LBNL-based 21st Century Accelerator team is depending upon the TOPS Center for advances in eigensolvers and in PDE-constrained optimization.

In addition, we will seek to be advised by several other applications groups within the DOE to strive for relevance and software technology transfer. An ANL/PNNL SciDAC team in quantum chemistry has incorporated PETSc in its past work and needs of scalable linear solvers and eigensolvers that we are interested in developing. The tokamak edge plasma SciDAC team from LLNL is in need of both time integration and scalable linear solvers, and is also interested in working with the TOPS Center. We will work with other DOE sponsored application groups whose simulations involve PDEs. Our current software packages have well over a hundred active DOE, university, and industrial users who have contributed invaluable suggestions and code back into our development cycles.

4.3 Internal Collaboration

The ETC director, Keyes, will coordinate technical integration, oversee the logistics of schedule and budget, and represent the ETC to DOE. A technical lead will be responsible for each of the topical areas: *time integrators and linear solvers* – Falgout, *nonlinear solvers and software integration and deployment* – Smith, *optimization* – Moré, *eigenanalysis and direct methods* – Ng, and *software performance* – Keyes. An advisory committee consisting of a representative from each of the tightly collaborating application centers, the AMR ETC and the TSTT ETC, plus the senior co-PIs in this ETC will provide technical oversight. There will be quarterly meetings of the technical advisory committee, as well as more frequent, informal meetings, via the Access Grid. Falgout, Smith, and Ng will also serve as "line managers" of the personnel charging to the ETC

at their respective laboratories. Since these three also share in the technical coordination, cross-laboratory responsibilities will be readily communicated.

Many interchanges of personnel are anticipated. Keyes presently spends approximately 25% of his time at LLNL (also convenient to Berkeley) under separate arrangements that are expected to continue into the early years of this Center, at least. Keyes also visits ANL, where he has graduate RAs in residence and pre-existing collaborations, approximately five times per year. Falgout plans a sabbatical at UC-Boulder during the first year of this Center, which will be concentrated on its goals. Other such arrangements, in both directions, are possible and will be encouraged and made feasible by this Center.

Some of the university work will be done by graduate research assistance at appropriate points in their careers. These RAs will also spend internships with laboratory partners.

Obligations that the participants have to the ETC include:

- Each research effort will either (a) be developing solution techniques that use a "hierarchy of approximations" to effect a nearly optimal complexity solution, or (b) be developing techniques that enable them. *For example, an ETC member would not be developing a new $O(n^2)$ algorithm or preparing optimizations for coding such algorithms. BUT a member may be developing a new parallel Gauss-Seidel smoother or an efficient implementation, for use inside a multigrid code.*

- Each researcher will interact on a regular basis with the other team members on algorithmic and software issues and hierarchy of approximation issues.

- Major sites will acquire and maintain or have access to an Access Grid node. These nodes will be available for both formal and informal ETC meetings. *For example, a AG node that required scheduling a month in advance would be unacceptable.*

- Software issues:

  (a) All software developed in the ETC will be usable from HYPRE or PETSc in a "natural" way.

  (b) All software developed in the ETC will leverage off other ETC software, i.e., it will use parts of other ETC software (when possible) to provide part of its core functionality rather than duplicating work. For example, if a new "grid-based" preconditioner is being developed it will use the grid-based solver interfaces defined by HYPRE, rather than another custom interface to the geometry. *N.B.: (a) and (b) do not imply that all ETC software developed must be part of PETSc or HYPRE. Stand alone software (for example a stand alone direct solver) may be developed and supported, but it will come with more than a "bolted-on after-thought" interface to the other ETC software.*

  (c) Each software component developer will respond in a timely manner to bug reports, usage questions and requests for additional functionality from other ETC members, as well as application users.

  (d) Except possibly for early prototypes, all software will be written to run in parallel on the current and next generation of machines using MPI.

  (e) Except possibly for early prototypes all software will be written to be portable to the common versions of Unix (Linux, solaris, AIX, IRIX) as well as Windows.

  (f) Coding groups will provide transparent access to their software on an on-going basis, for example, via the Web.

  (g) Coding groups will select their own (if any) revision control systems and coding standards.

When issues over these obligations arise they will be mediated by the ETC director. If not resolved to mutual satisfaction, they will be discussed by the director and four primary lab co-PIs and, if need be, resolved by a vote with one vote per lab.

References

[1] J. ABATE, S. BENSON, L. GRIGNON, P. HOVLAND, L. MCINNES, AND B. NORRIS, *Integrating automatic differentiation with object-oriented toolkits for high-performance scientific computing*, in Automatic Differentiation 2000: From Simulation to Optimization, G. Corliss, A. Griewank, C. Faure, L. Hascoët, and U. Naumann, eds., Springer, 2001. to appear.

[2] V. AKCELIK, I. EPANOMERITAKIS, AND O. GHATTAS, *Multiscale Gauss-Newton-Krylov methods for inverse wave propagation*, in Multiscale and Multiresolution Methods, Yosemite, CA, October 25-27, 2000.

[3] F. L. ALVARADO, A. POTHEN, AND R. SCHREIBER, *Highly parallel sparse triangular solution*, in Graph theory and sparse matrix computation, A. George, J. R. Gilbert, and J. W. Liu, eds., Springer-Verlag, New York, 1993, pp. 159–190.

[4] P. AMESTOY, T. A. DAVIS, AND I. S. DUFF, *An approximate minimum degree ordering algorithm*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 886–905.

[5] E. ANDERSON, Z. BAI, C. BISCHOF, S. BLACKFORD, J. DEMMEL, J. DONGARRA, J. DU CROZ, A. GREENBAUM, S. HAMMARLING, A. MCKENNEY, AND D. SORENSEN, *LAPACK Users' Guide, Release 3.0*, SIAM, Philadelphia, 1999. 407 pages.

[6] W. K. ANDERSON, W. D. GROPP, D. K. KAUSHIK, D. E. KEYES, AND B. F. SMITH, *Achieving high sustained performance in an unstructured mesh CFD application*, in Proceedings of SC'99, IEEE Computer Society, 1999. Gordon Bell Prize Award Paper in Special Category.

[7] J. ANTAKI, O. GHATTAS, G. BURGREEN, AND B. HE, *Computational flow optimization of rotory blood pump components*, Artificial Organs, 19 (1995), pp. 608–615.

[8] S. F. ASHBY AND R. D. FALGOUT, *A parallel multigrid preconditioned conjugate gradient algorithm for groundwater flow simulations*, Nuclear Science and Engineering, 124 (1996), pp. 145–159. Also available as LLNL Technical Report UCRL-JC-122359.

[9] O. AXELSSON, S. BRINKKEMPER, AND V. P. IL'IN, *On some versions of incomplete block-matrix factorization iterative methods*, Linear Algebra Appl., 58 (1984), pp. 3–15.

[10] O. AXELSSON AND V. EIJKHOUT, *The nested recursive two-level factorization method for nine-point difference matrices*, SIAM J. Sci. Comput., 12 (1991), pp. 1373–1400.

[11] M. BAERTSCHY, T. RESCIGNO, W. ISAACS, X. LI, AND C. MCCURDY, *Electron-impact ionization of atomic hydrogen*, Physical Review A, 63 022712 (2001).

[12] S. BALAY, W. D. GROPP, L. C. MCINNES, AND B. F. SMITH, *Efficient management of parallelism in object oriented numerical software libraries*, in Modern Software Tools in Scientific Computing, E. Arge, A. M. Bruaset, and H. P. Langtangen, eds., Birkhauser Press, 1997, pp. 163–202.

[13] ——, *PETSc 2.0 Users Manual*, Report ANL-95/11 - Revision 2.0.22, Argonne National Laboratory, Argonne, Illinois, 1998.

[14] C. BALDWIN, P. N. BROWN, R. D. FALGOUT, J. JONES, AND F. GRAZIANI, *Iterative linear solvers in a 2d radiation-hydrodynamics code: methods and performance*, Journal of Computational Physics, 154 (1999), pp. 1–40. Also available as LLNL technical report UCRL-JC-130933.

[15] R. E. BANK, *Analysis of a multi-level inverse iterative procedure for eigenvalue problems*, SIAM Journal on Numerical Analysis, 19 (1982), pp. 886–897.

[16] R. E. BANK AND R. K. SMITH, *The incomplete factorization multigraph algorithm*, SIAM J. Sci. Comput., 20 (1999), pp. 1349–1364.

[17] R. E. BANK AND C. WAGNER, *Multilevel ILU decomposition*, Numer. Math., 82 (1999), pp. 543–576.

[18] R. E. BANK AND J. XU, *A hierarchical basis multigrid method for unstructured grids*, in Fast Solvers for Flow Problems. Proceedings of the 10th GAMM-Seminar, Vieweg-Verlag, 1995.

[19] S. BENSON, L. C. MCINNES, AND J. J. MORÉ, *GPCG: A case study in the performance and scalability of optimization algorithms*, Preprint ANL/MCS-P768-0799, Argonne National Laboratory, 1999.

[20] ——, *TAO Users Manual*, Technical Manual ANL/MCS-TM-242, Argonne National Laboratory, 2000.

[21] J. BILMES, K. ASANOVIĆ, C. CHIN, AND J. DEMMEL, *Optimizing matrix multiply using PHiPAC: a portable, high-performance, ANSI C coding methodology*, in Proceedings of the International Conference on Supercomputing, Vienna, Austria, July 1997, ACM SIGARC. see `http://www.icsi.berkeley.edu/~bilmes/phipac`.

[22] J. BILMES, K. ASANOVIĆ, C.-W. CHIN, AND J. DEMMEL, *PHIPAC: A portable, high-performance, ANSI C coding methodology and its application to matrix multiply*. to appear in ICS; also available as LAPACK Working Note 111, http://www.netlib.org/lapack/lawns111, 1997.

[23] G. BIROS, *Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization, with Applications to Optimal Control of Vicous Fluids*, PhD thesis, Carnegie Mellon University, 2000.

[24] G. BIROS, *VELTISTO: A parallel library for PDE–constrained optimization*, 2000. Currently in beta testing.

[25] G. BIROS AND O. GHATTAS, *Parallel domain decomposition methods for optimal control of viscous incompressible flows*, in Parallel Computational Fluid Dynamics '99, A. Ecer and D. Emerson, eds., North Holland, 1999.

[26] ——, *Parallel Newton-Krylov algorithms for PDE–constrained optimization*, in Proceedings of SC99, Portland, Oregon, 1999.

[27] ——, *Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization. Part I: The Krylov-Schur solver*, 2000. See `http://www.cs.cmu.edu/~oghattas/papers/lnks/lnks1.pd`.

[28] ——, *Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization. Part II: The Lagrange-Newton solver, and its application to optimal control of steady viscous flows*, 2000. See `http://www.cs.cmu.edu/~oghattas/papers/lnks/lnks2.pdf`.

[29] S. BLACKFORD, G. CORLISS, J. DEMMEL, J. DONGARRA, I. DUFF, S. HAMMARLING, G. HENRY, M. HEROUX, C. HU, W. KAHAN, L. KAUFMAN, B. KEARFOTT, F. KROGH, X. LI, A. LUMSDAINE, Z. MAANY, A. PETITET, R. POZO, K. REMINGTON, W. WALSTER, C. WHALEY, AND J. W. V. GUDENBERG, *Basic Linear Algebra Subprograms Technical (BLAST) Forum Standard*, Intern. J. High Performance Comput. Appl., 15 (2001). to appear.

[30] A. BOOTEN, D. FOKKEMA, G. SLEIJPEN, AND H. VAN DER VORST, *Jacobi-Davidson methods for generalized MHD-eigenvalue problems*, Technical report Report NM-R9514, CWI, Dept of Num Math, Amsterdam, The Netherland, 1995.

[31] E. F. F. BOTTA, A. VAN DER PLOEG, AND F. W. WUBS, *Nested grids ILU-decomposition (NGILU)*, J. Comput. Appl. Math., 66 (1996), pp. 515–526.

[32] E. F. F. BOTTA AND F. W. WUBS, *Matrix Renumbering ILU: An effective algebraic multilevel ilu preconditioner for sparse matrices*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 1007–1026.

[33] R. L. BOWERS AND J. R. WILSON, *Numerical Modeling in Applied Physics and Astrophysics*, Jones and Bartlett, Boston, 1991.

[34] A. BRANDT, *Algebraic multigrid theory: The symmetric case*, in Preliminary Proceedings for the International Multigrid Conference, Copper Mountain, Colorado, April 1983.

[35] A. BRANDT, *Multigrid techniques: 1984 guide with applications to fluid dynamics*, GMD–Studien Nr. 85, Gesellschaft für Mathematik und Datenverarbeitung, St. Augustin, 1984.

[36] ——, *Algebraic multigrid theory: The symmetric case*, Appl. Math. Comput., 19 (1986), pp. 23–56.

[37] A. BRANDT, *General highly accurate algebraic coarsening*, ETNA, 10 (2000), pp. 1–20. Special issue on the Ninth Copper Mountain Conference on Multilevel Methods.

[38] A. BRANDT, S. F. MCCORMICK, AND J. RUGE, *Multigrid methods for differential eigenproblems*, SIAM J. Sci. Stat. Comput., 4 (1992), pp. 244–260.

[39] A. BRANDT, S. F. MCCORMICK, AND J. W. RUGE, *Algebraic multigrid (AMG) for automatic multigrid solutions with application to geodetic computations*. Report, Inst. for Computational Studies, Fort Collins, Colo., October 1982.

[40] ——, *Algebraic multigrid (AMG) for sparse matrix equations*, in Sparsity and Its Applications, D. J. Evans, ed., Cambridge University Press, Cambridge, 1984.

[41] M. BREZINA, A. J. CLEARY, R. D. FALGOUT, V. E. HENSON, J. E. JONES, T. A. MANTEUFFEL, S. F. MCCORMICK, AND J. W. RUGE, *Algebraic multigrid based on element interpolation (AMGe)*, SIAM J. Sci. Comput., 22 (2000), pp. 1570–1592. Also available as LLNL technical report UCRL-JC-131752.

[42] P. N. BROWN, R. D. FALGOUT, AND J. E. JONES, *Semicoarsening multigrid on distributed memory machines*, SIAM J. Sci. Comput., 21 (2000), pp. 1823–1834. Special issue on the Fifth Copper Mountain Conference on Iterative Methods. Also available as LLNL technical report UCRL-JC-130720.

[43] P. N. BROWN AND Y. SAAD, *Hybrid Krylov methods for nonlinear systems of equations*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 450–481.

[44] P. N. BROWN AND C. S. WOODWARD, *Preconditioning strategies for fully implicit radiation diffusion with material-energy transfer*, Center for Applied Scientific Computing UCRL-JC-139087, Lawrence Livermore National Laboratory, Livermore, CA, May 2000. To appear, SIAM J. Sci. Comput.

[45] G. D. BYRNE AND A. C. HINDMARSH, *PVODE, an ODE solver for parallel computers*, Int. J. High Perf. Comput. Appl., 13 (1999), pp. 354–365. Also available as LLNL Tech. Report UCRL-JC-132361.

[46] X.-C. CAI, W. D. GROPP, D. E. KEYES, R. G. MELVIN, AND D. P. YOUNG, *Parallel Newton-Krylov-Schwarz algorithms for the transonic full potential equation*, SIAM J. Sci. Comput., 19 (1998), pp. 246–265.

[47] X.-C. CAI AND D. E. KEYES, *Nonlinearly preconditioned inexact Newton algorithms*. in revision for SIAM J. Sci. Comput., July 2000.

[48] X.-C. CAI, D. E. KEYES, AND D. P. YOUNG, *A nonlinear additive Schwarz preconditioned inexact Newton method for shocked duct flow*. submitted to Proc. of the 13th International Conference on Domain Decomposition Methods, February 2001.

[49] G. F. CAREY, A. I. PEHLIVANOV, , AND P. S. VASSILEVSKI, *Least–squares mixed finite element methods for non–selfadjoint elliptic probelms: Ii. performance of block–ilu factorization methods*, SIAM J. Sci. Comput., 16 (1995), pp. 1126–1136.

[50] T. F. CHAN, S. D. MARGENOV, AND P. S. VASSILEVSKI, *Performance of block-ilu factorization preconditioners based on block-size reduction for 2d elasticity systems*, SIAM J. Sci. Comput., 18 (1997), pp. 1355–1366.

[51] T. F. CHAN AND P. S. VASSILEVSKI, *A framework for block–ilu factorizations using block–size reduction*, Mathematics of Computation, 64 (1995), pp. 129–156.

[52] E. CHOW AND P. S. VASSILEVSKI, *Improving the scalability of one-level preconditioning methods for solving large-scale linear systems*, in 16th IMACS World Congress 2000, Lausanne, Switzerland, August 21–25, 2000.

[53] R. L. CLAY, K. D. MISH, I. J. OTERO, L. M. TAYLOR, AND A. B. WILLIAMS, *An annotated reference guide to the finite-element interface (fei) specification: version 1.0*. Sandia National Laboratories report SAND99-8229, January 1999.

[54] A. J. CLEARY, R. D. FALGOUT, V. E. HENSON, AND J. E. JONES, *Coarse-grid selection for parallel algebraic multigrid*, in Proc. of the Fifth International Symposium on: Solving Irregularly Structured Problems in Parallel, vol. 1457 of Lecture Notes in Computer Science, New York, 1998, Springer–Verlag, pp. 104–115. Held at Lawrence Berkeley National Laboratory, Berkeley, CA, August 9–11, 1998. Also available as LLNL Technical Report UCRL-JC-130893.

[55] A. J. CLEARY, R. D. FALGOUT, V. E. HENSON, J. E. JONES, T. A. MANTEUFFEL, S. F. MCCORMICK, G. N. MIRANDA, AND J. W. RUGE, *Robustness and scalability of algebraic multigrid*, SIAM J. Sci. Comput., 21 (2000), pp. 1886–1908. Special issue on the Fifth Copper Mountain Conference on Iterative Methods. Also available as LLNL technical report UCRL-JC-130718.

[56] P. CONCUS, G. H. GOLUB, AND G. MEURANT, *Block preconditioning for the conjugate gradient method*, SIAM J. Sci. Stat. Comput., 6 (1985), pp. 309–332.

[57] J. CZYZYK, M. P. MESNIER, AND J. J. MORÉ, *The NEOS Server*, IEEE Computational Science & Engineering, 5 (1998), pp. 68–75.

[58] E. F. D'AZEVEDO, P. A. FORSYTH, AND W. P. TANG, *Ordering methods for preconditioned conjugate gradient methods applied to unstructured grid problems*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 944–961.

[59] J. W. DEMMEL, S. C. EISENSTAT, J. R. GILBERT, X. S. LI, AND J. W. H. LIU, *A supernodal approach to sparse partial pivoting*, SIAM J. Matrix Analysis and Applications, 20 (1999), pp. 720–755.

[60] J. W. DEMMEL, J. R. GILBERT, AND X. S. LI, *An asynchronous parallel supernodal algorithm for sparse gaussian elimination*, SIAM J. Matrix Analysis and Applications, 20 (1999), pp. 915–952.

[61] ——, *SuperLU Users' Guide*, Tech. Rep. LBNL-44289, Lawrence Berkeley National Laboratory, September 1999. Software is available at http://www.nersc.gov/∼xiaoye/SuperLU.

[62] J. DESCLOUX AND M. TOLLEY, *An accurate algorithm for computing the eigenvalues of a polygonal membrane*, Computer Methods in Applied Mechanics and Engineering, 39 (1983), pp. 37–53.

[63] I. DHILLON, G. FANN, AND B. PARLETT, *Application of a new algorithm for the symmetric eigenproblem to computational quantum chemistry*, in Proceedings of the Eighth SIAM Conference on Parallel Processing for Scientific Computing, SIAM, March 1997.

[64] I. S. DHILLON, *A New $O(n^2)$ Algorithm for the Symmetric Tridiagonal Eigenvalue/Eigenvector Problem*, PhD thesis, University of California, Berkeley, California, May 1997.

[65] ——, *Current inverse iteration software can fail*, BIT, 38 (1998), pp. 685–704.

[66] F. DOBRIAN, *Sparse factorization in computing systems with hierarchical storage*. PhD. thesis, 2001.

[67] F. DOBRIAN, G. KUMFERT, AND A. POTHEN, *Object-oriented design for sparse direct solvers*, in Modern Software Tools in Scientific Computing, H. P. Langtangen et al., ed., Springer Verlag, 2000, pp. 89–131. Lecture Notes in Computational Science and Engineering.

[68] J. DONGARRA, J. DU CROZ, I. S. DUFF, AND S. HAMMARLING, *A Set of Level 3 Basic Linear Algebra Subprograms*, ACM Trans. Math. Soft., 16 (1990), pp. 1–17.

[69] R. D. FALGOUT AND J. E. JONES, *Multigrid on massively parallel architectures*, in Multigrid Methods VI, E. Dick, K. Riemslagh, and J. Vierendeels, eds., vol. 14 of Lecture Notes in Computational Science and Engineering, Berlin, 2000, Springer, pp. 101–107. Proc. of the Sixth European Multigrid Conference held in Gent, Belgium, September 27-30, 1999. Also available as LLNL technical report UCRL-JC-133948.

[70] V. FERNANDO, B. PARLETT, AND I. DHILLON, *A way to find the most redundant equation in a tridiagonal system*. Berkeley Mathematics Dept. Preprint, 1995.

[71] O. GHATTAS AND J. BARK, *Large-scale SQP methods for optimization of Navier Stokes flows*, in Large-Scale Optimization with Applications. Part II: Optimal Control and Design, L. Biegler, T. Coleman, A. Conn, and F. Santosa, eds., Springer Verlag, Berlin, 1997, pp. 247–270.

[72] O. GHATTAS AND J.-H. BARK, *Optimal control of two- and three-dimensional incompressible Navier–Stokes flows*, Journal of Computational Physics, 136 (1997), pp. 231–244.

[73] O. GHATTAS AND X. LI, *Domain decomposition methods for sensitivity analysis of a nonlinear aeroelasticity problem*, International Journal of Computational Fluid Dynamics, 11 (1998), pp. 113–130.

[74] A. GRIEWANK, *Achieving logarithmic growth of temporal and spatial complexity in reverse mode automatic differentiation*, Optimization Methods and Software, 1 (1992), pp. 35–54.

[75] A. GRIEWANK, *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*, SIAM, 2000.

[76] W. GROPP AND J. J. MORÉ, *Optimization environments and the NEOS Server*, in Approximation Theory and Optimization, M. D. Buhmann and A. Iserles, eds., Cambridge University Press, 1997, pp. 167–182.

[77] W. D. GROPP, D. K. KAUSHIK, D. E. KEYES, AND B. F. SMITH, *High performance parallel implicit cfd*. Argonne Preprint ANL/MCS-P863-1200, to appear in Parallel Computing, 2001.

[78] W. D. GROPP, D. K. K. D. E. KEYES, AND B. F. SMITH, *Performance modeling and tuning of an unstructured mesh CFD application*, in Proceedings of SC2000, IEEE Computer Society, 2000.

[79] W. D. GROPP, L. C. MCINNES, M. D. TIDRIRI, AND D. E. KEYES, *Globalized Newton-Krylov-Schwarz algorithms and software for parallel implicit CFD*, Int. J. High Performance Computing Applications, 14 (2000), pp. 102–136.

[80] A. GUPTA, *Fast and effective algorithms for graph partitioning and sparse matrix ordering*, Tech. Rep. RC-20496, IBM Research Division, T.J. Watson Research Center, Yorktown Heights, NY 10598, 1996.

[81] W. HACKBUSCH, *On the computation of approximate eigenvalues and eigenvectors of elliptic operators by means of a multi-grid method*, SIAM Journal on Numerical Analysis, 16 (1979), pp. 201–215.

[82] B. HE, O. GHATTAS, AND J. ANTAKI, *Shape optimization of time-dependent Navier-Stokes flows*. AIAA 96-4187, 1996.

[83] M. T. HEATH AND P. RAGHAVAN, *A Cartesian nested dissection algorithm*, SIAM J. Matrix Anal. Appl., 16 (1995).

[84] ——, *Parallel sparse triangular solution*, in IMA Volumes in Mathematics and its Applications, R. Gulliver, M. T. Heath, R. Schreiber, and P. Bjorstad, eds., vol. 105, Springer-Verlag, 1998, pp. 289–306.

[85] B. HENDRICKSON AND E. ROTHBERG, *Improving the runtime and quality of nested dissection ordering*, tech. rep., Sandia National Laboratories, Albuquerque, NM 87185, 1996.

[86] V. E. HENSON AND P. S. VASSILEVSKI, *Element-free AMGe: General algorithms for computing interpolation weights in amg*, SIAM J. Sci. Comput., (2000). To appear. Also available as LLNL technical report UCRL-JC-139098.

[87] A. C. HINDMARSH AND A. G. TAYLOR, *User documentation for IDA, a differential-algebraic equation solver for sequential and parallel computers*, Tech. Rep. UCRL-MA-136910, Lawrence Livermore National Laboratory, 1999.

[88] P. D. HOVLAND, D. E. KEYES, L. C. MCINNES, AND W. SAMYONO, *Using automatic differentiation for second-order methods in PDE-constrained optimization*, in Automatic Differentiation 2000: From Simulation to Optimization, G. Corliss, A. Griewank, C. Faure, L. Hascoët, and U. Naumann, eds., Springer, 2001. to appear.

[89] D. HYSOM AND A. POTHEN, *A scalable parallel algorithm for incomplete factor preconditioners*, SIAM Journal on Scientific Computing, (2001), pp. 001–021. To appear. A preliminary version appeared in Supercomputing '99.

[90] E.-J. IM, *Automatic Optimization of Sparse Matrix - Vector Multiplication*, PhD thesis, University of California at Berkeley, May 2000.

[91] ——, *Papers on tuning sparse matrix vector multiplication*. `www.cs.berkeley.edu/~ejim`, 2000.

[92] E.-J. IM AND K. YELICK, *Optimizing sparse matrix vector multiplication on SMPs*, in Ninth SIAM Conference on Parallel Processing for Scientific Computing, San Antonio, TX, Mar. 1999.

[93] J. E. JONES AND P. S. VASSILEVSKI, *AMGe based on element agglomeration*, SIAM J. Sci. Comput., (2000). In press. Also available as LLNL technical report UCRL-JC-135441.

[94] J. E. JONES AND C. S. WOODWARD, *Newton-Krylov-multigrid solvers for large-scale, highly heterogeneous, variably saturated flow problems*, Center for Applied Scientific Computing UCRL-JC-138618, Lawrence Livermore National Laboratory, Livermore, CA, Apr. 2000. To appear, Advances in Water Resources.

[95] G. KARYPIS AND V. KUMAR, *Parallel multilevel graph partitioning*, Tech. Rep. 95-036, Department of Computer Science, University of Minnesota, Minneapolis, MN, May 1995.

[96] G. KARYPIS AND V. KUMAR, *A fast and high quality scheme for partitioning irregular graphs*, SIAM J. Scientific Computing, 20 (1999), pp. 359–392.

[97] D. E. KEYES, *How scalable is domain decomposition in practice?*, in Proceedings of the 11th International Conference on Domai n Decomposition Methods, C.-H. L. *et al.*, ed., Domain Decomposition Press, Bergen, 1999, pp. 286–297.

[98] ——, *Four horizons for enhancing the performance of parallel simulations based on partial differential equations*, in Proceedings of Europar 2000, Springer, 2000. Lecture Notes in Computer Science.

[99] D. A. KNOLL, P. R. MCHUGH, AND D. E. KEYES, *Domain decomposition methods applied to a system of convection-diffusion-reaction equations*, Zeitschr. Angew. Math. Mech., 76(Suppl. 1) (1996), pp. 235–238.

[100] R. B. LEHOUCQ, S. K. GRAY, D. H. ZHANG, AND J. C. LIGHT, *Vibrational eigenstates of four-atom molecules: A parallelization strategy employng the implicitly restarted lanczos method*, Computer Physics Comm., 109 (1998), pp. 15–26.

[101] X. S. LI AND J. W. DEMMEL, *A scalable sparse direct solver using static pivoting*, in Proceedings of the Ninth SIAM Conference on Parallel Processing for Scientific Computing, San Antonio, Texas, March 22–24 1999.

[102] J. W.-H. LIU, *Modification of the minimum degree algorithm by multiple elimination*, ACM Trans. Math. Software, 11 (1985), pp. 141–153.

[103] J. MANDEL, *Balancing domain decomposition*, Comm. Numer. Meth. Engrg., 9 (1993), pp. 233–241.

[104] J. MANDEL AND S. MCCORMICK, *A multi-level variational method for $Au = \lambda Bu$ on composite grids*, Journal of Computational Physics, 80 (1989), pp. 442–452.

[105] O. A. MARQUES AND Y.-H. SANEJOUAND, *Hinge-Bending Motion in Citrate Syntase Arising from Normal Modes Calculations*, Proteins, 23 (1995), pp. 557–560.

[106] R. M. MAXWELL, S. F. CARLE, C. S. WOODWARD, AND A. F. B. TOMPSON, *The impact of the geologic conceptual model on the prediction of human health risk from contaminated groundwater*. A poster presented (by Reed Maxwell) to the American Geophysical Union Winter Meeting, San Francisco, CA, 1998.

[107] S. F. MCCORMICK, *Multilevel Adaptive Methods for Partial Differential Equations*, vol. 6 of Frontiers in Applied Mathematics, SIAM Books, Philadelphia, 1989.

[108] ——, *Multi-level adaptive methods for elliptic eigenproblems: A two-level convergence theory*, SIAM Journal on Numerical Analysis, 31 (1994), pp. 1731–1745.

[109] P. R. MCHUGH, D. A. KNOLL, AND D. E. KEYES, *Application of a schwarz-preconditioned newton-krylov algorithm to a low-speed reacting flow problem*, AIAA J., 36 (1998), pp. 290–292.

[110] H. D. MITTELMANN AND H. MAURER, *Solving elliptic control problems with interior point and SQP methods: Control and state constraints*, J. Comp. Appl. Math., 120 (2000), pp. 175–195.

[111] E. NG AND B. W. PEYTON, *Block sparse Cholesky algorithms on advanced uniprocessor computers*, SIAM J. Sci. Comput., 14 (1993), pp. 1034–1056.

[112] E. NG AND P. RAGHAVAN, *The performance of greedy ordering heuristics for sparse Cholesky factorization*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 902–914.

[113] E. G. NG AND B. W. PEYTON, *A supernodal Cholesky factorization algorithm for shared-memory multiprocessors*, SIAM J. Sci. Comput., 14 (1993), pp. 761–769.

[114] C. E. OROZCO AND O. GHATTAS, *Infeasible path optimal design methods, with application to aerodynamic shape optimization*, AIAA Journal, 34 (1996), pp. 217–224.

[115] ——, *A reduced SAND method for optimal design of nonlinear structures*, International Journal for Numerical Methods in Engineering, 40 (1997), pp. 2759–2774.

[116] B. PARLETT AND I. DHILLON, *Fernando's solution to Wilkinson's problem: An application of double factorization*, Lin. Alg. Appl., 267 (1997), pp. 247–279.

[117] G. C. POMRANING, *The Equations of Radiation Hydrodynamics*, Pergamon, New York, 1973.

[118] P. RAGHAVAN, *Parallel ordering using edge contraction*, Parallel Computing, 23 (1997), pp. 1045–1067.

[119] ——, *Efficient parallel triangular solution with selective inversion*, Parallel Processing Letters, 8 (1998), pp. 29–40.

[120] ——, *DSCPACK: A domain-separator Cholesky package*, 2000. Software for solving sparse linear systems on multiprocessors and NOWs using C and MPI. Package has a two stage parallel nested dissection, higher-level BLAS, fast repeated solves and an easy to use parallel interface.

[121] A. REUSKEN, *A multigrid method based on incomplete Gaussian elimination*, Num. Lin. Alg. Appl., 3 (1996), pp. 369–390.

[122] J. W. RUGE AND K. STÜBEN, *Efficient solution of finite difference and finite element equations by algebraic multigrid (AMG)*, in Multigrid Methods for Integral and Differential Equations, D. J. Paddon and H. Holstein, eds., The Institute of Mathematics and its Applications Conference Series, Clarendon Press, Oxford, 1985, pp. 169–212.

[123] ——, *Algebraic multigrid (AMG)*, in Multigrid Methods, S. F. McCormick, ed., vol. 3 of Frontiers in Applied Mathematics, SIAM, Philadelphia, PA, 1987, pp. 73–130.

[124] R. SHEPARD, A. F. WAGNER, J. TILSON, AND M. MINKOFF, *The subspace projected approximate matrix (SPAM) modification of the Davidson method*, technical report, Argonne National Laboratory, 2000. Submitted to Journal of Computational Physics.

[125] G. L. G. SLEIJPEN AND H. A. VAN DER VORST, *A Jacobi-Davidson iteration method for linear eigenvalue problems*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 401–425.

[126] D. C. SORENSEN, *Truncated QZ methods for large scale generalized eigenvalue problems*, ETNA, 7 (1998), pp. 141–162.

[127] D. C. SORENSEN AND C. YANG, *A truncated RQ-iteration for large scale eigenvalue calculations*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 1045–1073.

[128] K. STÜBEN, *Algebraic multigrid (AMG): experiences and comparisons*, Appl. Math. Comput., 13 (1983), pp. 419–452.

[129] K. STÜBEN, U. TROTTENBERG, AND K. WITSCH, *Software development based on multigrid techniques*, in Proc. IFIP–Conference on PDE Software, Modules, Interfaces and Systems, B. Enquist and T. Smedsaas, eds., Sweden, 1983, Söderköping.

[130] A. G. TAYLOR AND A. C. HINDMARSH, *User documentation for KINSOL, a nonlinear solver for sequential and parallel computers*, Tech. Rep. UCRL-ID-131185, Lawrence Livermore National Laboratory, 1998.

[131] H. J. J. VAN DAM, J. H. VAN LENTHE, G. L. G. SLEIJPEN, AND H. A. VAN DER VORST, *An improvement of Davidson's iteration: Applications to MRCI and MRCEPA calculations*, J. Comput. Chemistry, 17 (1996), pp. 267–272.

[132] P. VANĚK, J. MANDEL, AND M. BREZINA, *Algebraic multigrid based on smoothed aggregation for second and fourth order problems*, Computing, 56 (1996), pp. 179–196.

[133] D. W. VASCO, L. R. JOHNSON, AND O. A. MARQUES, *Global Earth Sctructure: Inference and Assessment*, Geophysical Journal International, 137 (1999), pp. 381–407.

[134] R. VUDUC, J. DEMMEL, AND J. BILMES, *Statistical modeling of feedback data in an automatic tuning system*, in MICRO-33: Third ACM Workshop on Feedback-Directed Dynamic Optimization, December 2000.

[135] R. C. WHALEY AND J. DONGARRA, *The ATLAS WWW home page*. `http://www.netlib.org/atlas/`.

[136] D. WHITE AND J. KONING, *Efficient solution of large-scale electromagnetic eigenvalue problems using the implicitly restarted Arnoldi method*, in 9th Biennial IEEE Conference on Electromagnetic Field Computation, 2000.

[137] C. S. WOODWARD, *A Newton-Krylov-multigrid solver for variably saturated flow problems*, in Proc. of the Twelfth International Conference on Computational Methods in Water Resources, vol. 2, Southampton, 1998, Computational Mechanics Publications, pp. 609–616. Held in Crete, Greece, June, 1998. Also available as LLNL Technical Report UCRL-JC-129371.

[138] C. YANG AND W. F. LAWKINS, *A multi-level method for large-scale eigenvalue calculation*, Technical report ORNL TM-2000/43, Computer Science & Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, TN, 2000.

[139] C. YANG, B. W. PEYTON, D. W. NOID, B. G. SUMPTER, AND R. E. TUZUN, *Large-scale normal coordinate analysis for molecular structures*, SIAM J. Sci. Comp., To appear (2001).

[140] C. YANG, P. RAGHAVAN, L. ARROWOOD, D. W. NOID, B. G. SUMPTER, AND R. E. TUZUN, *Large-scale normal coordinate analysis on distributed memory parallel systems*, technical report, NERSC, Berkeley, CA, 2001. In preparation.

5 Budget

We request an initial base budget of approximately $4M per year to achieve our objectives of research, development, deployment, and support. These resources concentrate at the laboratories, where there are specific code maintenance responsibilities.

Each of the nine participating organizations presents its own budget and will be considered as a prime contractor in this common proposal. Budget justifications accompany individual organizational budgets. In particular, brief synopses of which personnel are attached to which projects are furnished therein. (Together, these serve as a complementary index to the listings of personnel by topic in Section 3.)

Most of the universities participating in this Center request inflationary increases at constant level of effort in the out-years of activity.

The laboratories participating in this Center request both inflationary increases for their base budget personnel through the out-years of the proposal, and also additional personnel increments in Years 2 and 3. With anticipated growth in demand for TOPS software, as it acquires a broader user community, we anticipate the need to train new staff in its use and maintenance, and new consulting responsibilities. These will be the prime applications of the incremental resources at LLNL and LBNL.

In addition to supporting its own suite of codes for a broader audience, Argonne will further develop their C-based Automatic Differentiation software for PDE-oriented uses, as mentioned in this proposal.

In the event that the additional inflationary resources do not become available, the levels of effort associated with the Center will decrease accordingly, since almost all of the support is allocated to people. These reductions would not be simply "across the board," although we do not anticipating cutting out any institutional partner altogether. First, we would seek to identify leveraging possibilities that would preserve the "comprehensive" coverage of this PDE-oriented software suite. Second, we would curtail new research under this Center and concentrate on hardening and supporting already existing code investments. Third, we would regretfully sacrifice code projects in the areas that are auxiliary to our PDE-oriented core of nonlinear and linear solvers: the optimization and eigenanalysis efforts. Nevertheless, we emphasize that these code capabilities belong together with the PDE solvers, since users of the latter should have transparent access to the former, as well.

In the event that the incremental resources do not become available, level of external support will not rise to match the anticipated demand. Nevertheless, no eager "client" group will be turned away. Rather, clients will be expected to contribute more of their own personnel to enable fruitful collaborations, and the rate of technology transfer will be somewhat slower.

6 Other Support

Current and pending support for each named senior investigator is detailed in the institutional financials.

7 Biographical Sketches

TOPS Center participants are drawn from three DOE labs and six universities, as listed alphabetically by organization below. Biosketches from the lead investigator at each organization and certain thrust leaders are also included in this section, alphabetically by individual.

Argonne National Laboratory
Lead co-PI: Barry Smith
Co-PIs: Steve Benson, Jorge Moré, Todd Munson

Lawrence Berkeley National Laboratory
Lead co-PI: Esmond Ng
Co-PIs: Parry Husbands, Xiaoye Sherry Li, Osni Marques

Lawrence Livermore National Laboratory
Lead co-PI: Rob Falgout
Co-PIs: Edmond Chow, Steve Lee, Panayot Vassilevski, Carol Woodward

Carnegie Mellon University
Lead co-PI: Omar Ghattas

New York University
Lead co-PI: Olof Widlund
Co-PI: George Biros

Old Dominion University
Lead co-PI: David Keyes
Co-PI: Alex Pothen

University of California-Berkeley
Lead co-PI: James Demmel

University of Colorado-Boulder
Lead co-PI: Steve McCormick
Co-PIs: Xiao-Chuan Cai, Tom Manteuffel

University of Tennessee
Lead co-PI: Jack Dongarra